


```
CCCCCCCC 000000 NN NN TTTTTTTTTT EEEEEEEEE NN NN TTTTTTTTTT SSSSSSSS
CCCCCCCC 000000 NN NN TTTTTTTTTT EEEEEEEEE NN NN TTTTTTTTTT SSSSSSSS
CC 00 00 NN NN TT EE NN NN TT SS
CC 00 00 NN NN TT EE NN NN TT SS
CC 00 00 NNNN NN TT EE NNNN NN TT SS
CC 00 00 NNNN NN TT EE NNNN NN TT SS
CC 00 00 NN NN NN TT EEEEEEE NN NN TT SSSSSS
CC 00 00 NN NN NN TT EEEEEEE NN NN TT SSSSSS
CC 00 00 NN NNNN TT EE NN NNNN TT SS
CC 00 00 NN NNNN TT EE NN NNNN TT SS
CC 00 00 NN NN TT EE NN NN TT SS
CC 00 00 NN NN TT EE NN NN TT SS
CCCCCCCC 000000 NN NN TT EEEEEEEEE NN NN TT SSSSSSSS
CCCCCCCC 000000 NN NN TT EEEEEEEEE NN NN TT SSSSSSSS

LL 111111 SSSSSSSS
LL 111111 SSSSSSSS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SSSSSS
LL 11 SSSSSS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS
```

```
0001 0 %TITLE 'CONTENTS - Table of Contents formatter program'
0002 0 !<BLF/NOFORMAT>
0003 0
0004 0 MODULE contents (IDENT = 'V04-000'
0005 0 %BLISS32[, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE)]
0006 0 ) =
0007 1 BEGIN
0008 1
0009 1 !<BLF/FORMAT>
0010 1 !<BLF/LOWERCASE_USER>
0011 1 !<BLF/UPPERCASE_KEY>
0012 1 !<BLF/MACRO>
0013 1
0014 1
0015 1 *****
0016 1 *
0017 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0018 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0019 1 * ALL RIGHTS RESERVED.
0020 1 *
0021 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0022 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0023 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0024 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0025 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0026 1 * TRANSFERRED.
0027 1 *
0028 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0029 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0030 1 * CORPORATION.
0031 1 *
0032 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0033 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0034 1 *
0035 1 *
0036 1 *****
0037 1
0038 1
0039 1 ++
0040 1 FACILITY:
0041 1 DSR (Digital Standard RUNOFF) /DSRPLUS DSRTOC/CONTENTS Utility
0042 1
0043 1 ABSTRACT:
0044 1 Read and process a RUNOFF binary table of contents file.
0045 1
0046 1 ENVIRONMENT: Transportable
0047 1
0048 1 AUTHOR: JPK
0049 1
0050 1 CREATION DATE: March 1982
0051 1
0052 1 MODIFIED BY:
0053 1
0054 1 013 JPK00011 20-May-1983
0055 1 Modified CONTENTS to check status of $XPO_PARSE_SPEC to avoid
0056 1 XPORT error messages.
0057 1
```

58	0058	1	012	JPK00010	05-May-1983	Modified CONTENTS to change text for index record group to all caps. Modified CAPTION to change the number of blank lines after the entity title line.
59	0059	1				
60	0060	1				
61	0061	1				
62	0062	1				
63	0063	1	011	JPK00009	24-Mar-1983	Modified CNTT20 to support new command line syntax. Modified CONTENTS adding routines PROCESS_PAGE, PROCESS_ENTITY_INFO and PROCESS_ENTITY_TXT to remove code from routine TOC so that CONTENTS will compile on TOPS-20. Modified CNTVMSREQ to remove conditional require of RNODEF.
64	0064	1				
65	0065	1				
66	0066	1				
67	0067	1				
68	0068	1				
69	0069	1				
70	0070	1	010	JPK00008	09-Mar-1983	Modified CONTENTS and CAPTION to support new BRN formats, support SEND CONTENTS, /DOUBLE_SPACE, page numbered chapters, guarantee space after section number and to write new prologue and epilog for RUNOFF output. Modified FORMAT to quote only the RUNOFF flags used by CONTENTS. Modified CNTVMS to fix default for /DOUBLE_SPACE and do more value checking.
71	0071	1				
72	0072	1				
73	0073	1				
74	0074	1				
75	0075	1				
76	0076	1				
77	0077	1				
78	0078	1				
79	0079	1	009	JPK00007	14-Feb-1983	Global edit of all sources for CONTENTS/DSRTOC: - module names are now consistent with file names - copyright dates have been updated - facility names have been updated - revision history was updated to be consistent with DSR/DSRPLUS
80	0080	1				
81	0081	1				
82	0082	1				
83	0083	1				
84	0084	1				
85	0085	1				
86	0086	1	008	JPK00006	14-Feb-1983	Modified CNTVMS, CONTENTS, FORMAT and CNTVMSMSG to generate error messages for DSRTOC or CONTENTS depending on the compiletime variant for DSRPLUS (/VARIANT:8192)
87	0087	1				
88	0088	1				
89	0089	1				
90	0090	1				
91	0091	1	007	JPK00005	14-Feb-1983	Changed the names of the temporary files used to store .SEND TOC, figure, table and example captions from XPORT.TMP;1, :2, :3, :4 to OCNTST.TMP, OCNTFG.TMP, OCNTTB.TMP, and OCNTX.TMP respectively. Delete the output file if /NOOUTPUT was specified on the command line. Module CONTENTS modified.
92	0092	1				
93	0093	1				
94	0094	1				
95	0095	1				
96	0096	1				
97	0097	1				
98	0098	1	006	JPK00004	11-Feb-1983	Changed the global variable name INDENT to LINE_INDENT in modules CONTENTS, CAPTION, FORMAT and GBLDCL. Removed declarations of PDENTS in modules CNTVMS, CONTENTS, and CAPTION and replaced with a module wide BIND using the new name INDENTS. Changed handling of INDENTS [1]. It no longer represents the sum of the chapter and title indents.
99	0099	1				
100	0100	1				
101	0101	1				
102	0102	1				
103	0103	1				
104	0104	1				
105	0105	1				
106	0106	1				
107	0107	1	005	KFA00003	04-Feb-1983	Modified format of header level one to not include ".0" in the counter. Change was in OUTSEC in module CONTENTS.
108	0108	1				
109	0109	1				
110	0110	1				
111	0111	1	004	KFA00002	14-Oct-1982	Modified CNTVMS, CONTENTS, CAPTION and CNTCLI.REQ to handle new syntax for /INDENT qualifier. All indents are now stored in the vector CMDBLK [CONTENTSSAG_HL_INDENT []],
112	0112	1				
113	0113	1				
114	0114	1				

115	0115	1	where 0 = chapter indent, 1 = header level one indent, etc.	
116	0116	1		
117	0117	1	003 KFA00001 13-Oct-1982	
118	0118	1	Modified CONTENTS and CAPTION to allow null captions for	
119	0119	1	headers, examples, figures, & tables.	
120	0120	1		
121	0121	1	002 JPK00001 17-Sep-1982	
122	0122	1	Modified CONTENTS for TOPS-20. This included splitting routine	
123	0123	1	toc () into toc () and tocfin () so that it can be compiled on	
124	0124	1	the 20, and calling \$XPO_GET_MEM and \$XPO_FREE_MEM to	
125	0125	1	temporarily store the prefix and suffix strings for HEADER	
126	0126	1	EXAMPLE, FIGURE and TABLE.	
127	0127	1		
128	0128	1	--	
129	0129	1		
130	0130	1		
131	0131	1	TABLE OF CONTENTS:	
132	0132	1		
133	0133	1		
134	0134	1	FORWARD ROUTINE	
135	0135	1	toc,	Main routine
136	0136	1	tocini,	Do initialization
137	0137	1	process_page : NOVALUE,	Process page reference record
138	0138	1	process_entity_info : NOVALUE,	Process figure/table/example info record
139	0139	1	process_entity_txt : NOVALUE,	Process figure/table/example text record
140	0140	1	tocfin,	Complete processing and clean up
141	0141	1	vfyap,	Check caption style
142	0142	1	outsec : NOVALUE;	Generate section number
143	0143	1		
144	0144	1		
145	0145	1	INCLUDE FILES:	
146	0146	1		
147	0147	1		
148	0148	1	LIBRARY 'NXPORT:XPORT';	
149	0149	1		
150	0150	1	%IF %BLISS (BLISS36)	
151	0151	1	%THEN	
152	0152	1		
153	0153	1	REQUIRE 'REQ:RNODEF';	
154	0154	1		
155	0155	1	%FI	
156	0156	1		
157	0157	1	%IF %BLISS (BLISS32)	
158	0158	1	%THEN	
159	0159	1		
160	0160	1	REQUIRE 'REQ:CNVMSREQ';	! VMS error messages
161	0400	1		
162	0401	1	%FI	
163	0402	1		
164	0403	1	REQUIRE 'REQ:BRNRTY';	! Binary RUNOFF file formats
165	0483	1		
166	0484	1	REQUIRE 'REQ:PAGEN';	! Page reference format
167	0565	1		
168	0566	1	REQUIRE 'REQ:TOCRTY';	! Table of Contents file formats
169	0676	1		
170	0677	1	REQUIRE 'REQ:CNTCLI';	! Command line information block formats
171	0805	1		

```

172 0806 1  |
173 0807 1  | MACROS:
174 0808 1  |
175 0809 1  |
176 0810 1  | MACRO
177 0811 1  |
178 0812 1  |   Write a character to output line
179 0813 1  |
180 M 0814 1  |   write_char (ch) [] =
181 M 0815 1  |   BEGIN
182 M 0816 1  |   CH$WCHAR_A (ch, lp);
183 M 0817 1  |   intlin = .intlin + 1;
184 M 0818 1  |
185 M 0819 1  |   %IF NOT %NULL (%REMAINING)
186 M 0820 1  |   %THEN
187 M 0821 1  |       extlin = .extlin + 1;
188 M 0822 1  |   %FI
189 M 0823 1  |
190 M 0824 1  |   END
191 0825 1  |
192 0826 1  |
193 0827 1  |   Write a text literal to the output line
194 0828 1  |
195 M 0829 1  |   literal_text (str) =
196 M 0830 1  |   BEGIN
197 M 0831 1  |   CH$MOVE (%CHARCOUNT (str), CH$PTR (UPLIT (str)), .lp);
198 M 0832 1  |   lp = CH$PLUS (.lp, %CHARCOUNT (str));
199 M 0833 1  |   intlin = .intlin + %CHARCOUNT (str);
200 M 0834 1  |   END
201 0835 1  |
202 0836 1  |
203 0837 1  |   Pad the output line with blanks
204 0838 1  |
205 M 0839 1  |   pad (n_blanks) =
206 M 0840 1  |   BEGIN
207 M 0841 1  |
208 M 0842 1  |   IF n_blanks GTR 0
209 M 0843 1  |   THEN
210 M 0844 1  |       BEGIN
211 M 0845 1  |       CH$FILL (%C' ', n_blanks, .lp);
212 M 0846 1  |       lp = CH$PLUS (.lp, n_blanks);
213 M 0847 1  |       intlin = .intlin + n_blanks;
214 M 0848 1  |       extlin = .extlin + n_blanks;
215 M 0849 1  |       END;
216 M 0850 1  |
217 M 0851 1  |   END
218 0852 1  |
219 0853 1  |
220 0854 1  |   Clear the text lines being built up.
221 0855 1  |
222 M 0856 1  |   clr_line (_) =
223 M 0857 1  |   BEGIN
224 M 0858 1  |   lp = CH$PTR (line);
225 M 0859 1  |   intlin = 0;
226 M 0860 1  |   extlin = 0;
227 M 0861 1  |   END
228 0862 1  |

```

```

229 0863 1      |
230 0864 1      | Insert specified character sequence into file, as is.
231 0865 1      |
232 M 0866 1      | put (str) =
233 M 0867 1      | BEGIN
234 M 0868 1      |   $str_copy (string = str, target = tmpstr);
235 M 0869 1      |   chrout = .chrout + .tmpstr [str$length];
236 M 0870 1      |   $xpo_put (iob = tocoob, string = tmpstr);
237 M 0871 1      |
238 M 0872 1      |   For TMS output, split the output file if it gets too large
239 M 0873 1      |
240 M 0874 1      |
241 M 0875 1      |   IF .cmdblk [contents$v_tms11] THEN split ();
242 M 0876 1      |
243 M 0877 1      |   END
244 M 0878 1      | %;
245 M 0879 1      |
246 M 0880 1      |
247 M 0881 1      | EQUATED SYMBOLS:
248 M 0882 1      |
249 M 0883 1      |
250 M 0884 1      | LITERAL
251 M 0885 1      |   true = 1,
252 M 0886 1      |   false = 0,
253 M 0887 1      |   new_captions = 1,
254 M 0888 1      |   old_captions = -1,
255 M 0889 1      |   tms_line_length = 600,
256 M 0890 1      |   infinity = 1*(%BPVAL - 1) - 1;
257 M 0891 1      |
258 M 0892 1      |
259 M 0893 1      | OWN STORAGE:
260 M 0894 1      |
261 M 0895 1      |
262 M 0896 1      | OWN
263 M 0897 1      |   tociob : $xpo_iob (),
264 M 0898 1      |   file_type,
265 M 0899 1      |   fgioB : $xpo_iob (),
266 M 0900 1      |   tbiob : $xpo_iob (),
267 M 0901 1      |   exiob : $xpo_iob (),
268 M 0902 1      |   hlcnt,
269 M 0903 1      |   lasthl,
270 M 0904 1      |   toc_page : INITIAL (false),
271 M 0905 1      |   captyp,
272 M 0906 1      |   capmsg,
273 M 0907 1      |   errcnt;
274 M 0908 1      |
275 M 0909 1      |
276 M 0910 1      | EXTERNAL REFERENCES:
277 M 0911 1      |
278 M 0912 1      |
279 M 0913 1      | EXTERNAL
280 M 0914 1      |   cmdblk : $contents_cmd,
281 M 0915 1      |   tocoob : $xpo_iob (),
282 M 0916 1      |   stciob : $xpo_iob (),
283 M 0917 1      |   cntvrl,
284 M 0918 1      |   cntvrp,
285 M 0919 1      |   chrout,

```

```

! Using HEADER FIGURE, etc.
! Using SEND TOC for captions

! IOB for binary index file.
! Input file type
! IOB for HEADER FIGURES
! IOB for HEADER TABLES
! IOB for HEADER EXAMPLES
! Maximum number of header levels
! previous "n"
! True if a CHAPTER should have a page reference
! To tell if SEND TOC or HEADER x was used
! True if incompatible captions and message was written
! Count of number of CONFUSED messages.

```

```

! Command line information block
! IOB for the resulting .RNT file
! IOB for send toc temp file
! Length of version number string
! CHSPTR to version number string
! Number of characters written to output file

```

286	0920	1	tmpstr : \$str_descriptor (),	For temporary strings
287	0921	1	txtlen,	Length of input text.
288	0922	1	hl_n,	'n' from latest .HL n command
289	0923	1	hl_list : VECTOR [100],	Header level counters
290	0924	1	hldsp : VECTOR [100],	Header level display characteristics
291	0925	1	major,	Major record type code
292	0926	1	minor,	Minor record type code
293	0927	1	pagen : page_definition,	Current page number.
294	0928	1	nxtstc,	Next SEND TOC 'n' value to process
295	0929	1	rno_ex : \$str_descriptor (),	String constant 'EXAMPLES'
296	0930	1	rno_fg : \$str_descriptor (),	String constant 'FIGURES'
297	0931	1	rno_tb : \$str_descriptor (),	String constant 'TABLES'
298	0932	1	tms_ex : \$str_descriptor (),	String constant 'Examples'
299	0933	1	tms_fg : \$str_descriptor (),	String constant 'Figures'
300	0934	1	tms_tb : \$str_descriptor (),	String constant 'Tables'
301	0935	1	prestr : \$str_descriptor (),	Caption prefix strings
302	0936	1	sfxstr : \$str_descriptor (),	Caption suffix strings
303	0937	1	lp,	CHSPTR along line being built up
304	0938	1	intlin,	Number of characters needed to represent text
305	0939	1	extlin,	Number of resulting print positions
306	0940	1	line : VECTOR [CH\$ALLOCATION (10000)],	Buffer in which line is being built up.
307	0941	1	lenpag,	Number of characters in the converted page number.
308	0942	1	txtpag : VECTOR [CH\$ALLOCATION (50)],	The text (lots of room)
309	0943	1	ptrpag,	A CHSPTR (working pointer)
310	0944	1	lensct,	Number of characters in the converted section number
311	0945	1	txtsct : VECTOR [CH\$ALLOCATION (150)],	The text (lots of room)
312	0946	1	ptrsct,	A CHSPTR (working pointer)
313	0947	1	rmargin,	Used by ENDWRD for controlling filling lines.
314	0948	1	wrap,	Wrap long lines around to here.
315	0949	1	line_indent;	Assume this standard indentation before the text.
316	0950	1		
317	0951	1	BIND	
318	0952	1	indents = cmdblk [contents\$ag_hl_indent] : VECTOR;	
319	0953	1		
320	0954	1	EXTERNAL ROUTINE	
321	0955	1	fmttxt : NOVALUE,	Generate formatted text
322	0956	1	insref : NOVALUE,	Insert a page reference
323	0957	1	hdr_x : NOVALUE,	Process .HEADER x Captions
324	0958	1	stcHdr : NOVALUE,	Process .SEND TOC n Captions
325	0959	1	insrnt : NOVALUE,	Insert text directly into RNT
326	0960	1	split : NOVALUE,	Start new output file for TMS
327	0961	1	pacbas,	Convert a number to ASCII
328	0962	1	pacpag,	Convert a PAGEN to a page number
329	0963	1	pacsec,	Format a section number
330	0964	1	pacxxx;	Convert number to ASCII using display codes
331	0965	1		
332	0966	1	!IF !BLISS (BLISS32)	
333	0967	1	!THEN	
334	0968	1		
335	0969	1	EXTERNAL ROUTINE	
336	0970	1	open_error;	! Action routine to handle file open failures
337	0971	1		
338	0972	1	!FI	


```

340 0973 1 XSBTTL 'TOC - Transportable entry point'
341 0974 1 GLOBAL ROUTINE toc =
342 0975 1 ++
343 0976 1 FUNCTIONAL DESCRIPTION:
344 0977 1
345 0978 1     Reads a binary table of contents file, as generated by RUNOFF
346 0979 1     and uses it to generate a .RNT file.
347 0980 1
348 0981 1 FORMAL PARAMETERS:
349 0982 1
350 0983 1     NONE
351 0984 1
352 0985 1 IMPLICIT INPUTS:
353 0986 1
354 0987 1     Implicit is the order in which records are written, and the
355 0988 1     format of the file. The format of the file is adequately
356 0989 1     documented by the code. It will not be further explained.
357 0990 1
358 0991 1     The order of information in the file is not obvious, although
359 0992 1     its use shows what is available when.
360 0993 1
361 0994 1     Briefly, there is an entry in the file (MAJ_PAGE) for every page
362 0995 1     in the document, regardless of whether or not that page is
363 0996 1     mentioned in the table of contents. Further, the page number is
364 0997 1     guaranteed to occur before you get some text associated with
365 0998 1     that page. So, if header level text is found, the page number
366 0999 1     is already correct.
367 1000 1
368 1001 1     Similarly, header level numbers and display information are
369 1002 1     always written before the header level text to which it is
370 1003 1     applicable. Note that the header level information is *not*
371 1004 1     reset in the file at the start of a chapter; you have to wait
372 1005 1     for the next header level number record to be input to be sure
373 1006 1     where you are. The page number and header level information is
374 1007 1     as valid for text from the .SEND command as it is for text from
375 1008 1     a .HL command.
376 1009 1
377 1010 1 IMPLICIT OUTPUTS:
378 1011 1
379 1012 1     NONE
380 1013 1
381 1014 1 ROUTINE VALUE:
382 1015 1 COMPLETION CODES:
383 1016 1
384 1017 1     NONE
385 1018 1
386 1019 1 SIDE EFFECTS:
387 1020 1
388 1021 1     NONE
389 1022 1
390 1023 1 --
391 1024 1
392 1025 2 BEGIN
393 1026 2
394 1027 2     Do initialization
395 1028 2
396 1029 2 IF NOT tocini () THEN RETURN false;

```

```

397 1030 2
398 1031 2
399 1032 2 Read record-groups until end of file. Process each record group
400 1033 2 in turn. Return at end-of-file.
401 1034 2
402 1035 2 WHILE $xpo_get (iob = tociob, fullwords = 2) EQL xpo$_normal DO
403 1036 3 BEGIN
404 1037 3
405 1038 3 BIND
406 1039 3 codes = .toclob [iob$a_data] : VECTOR;
407 1040 3
408 1041 4 IF (.file_type EQL brn_file) AND (.codes [0] NEQ brn_contents)
409 1042 3 THEN
410 1043 4 BEGIN
411 1044 4
412 1045 4 Not a contents record group. Skip it.
413 1046 4
414 1047 4 $xpo_get (iob = tociob, fullwords = .codes [1]);
415 1048 4 END
416 1049 3 ELSE
417 1050 4 BEGIN
418 1051 4
419 1052 4 A contents record group or not a .BRN file
420 1053 4
421 1054 4 IF .file_type EQL brn_file THEN $xpo_get (iob = tociob, fullwords = 2);
422 1055 4
423 1056 4 major = .codes [0];
424 1057 4 minor = .codes [1];
425 1058 4
426 1059 4 SELECTONE true OF
427 1060 4 SET
428 1061 4
429 1062 4 [.major EQL maj_new_toc] :
430 1063 5 BEGIN
431 1064 5
432 1065 5 File identification record in a BRN file
433 1066 5
434 1067 6 IF (.file_type NEQ brn_file)
435 1068 6 OR (.minor NEQ toc_format)
436 1069 5 THEN
437 1070 6 BEGIN
438 1071 6
439 1072 6 Report bad file format and quit
440 1073 6
441 1074 6
442 L 1075 6 %IF %BLISS (BLISS32)
443 1076 6 %THEN
444 1077 6 SIGNAL (contents$_invinput, 1, tociob [iob$t_resultant]);
445 U 1078 6 %ELSE
446 U 1079 6 $xpo_put_msg (severity = warning,
447 U 1080 6 string = $str_concat ('invalid file format ',
448 U 1081 6 tociob [iob$t_resultant], ''));
449 1082 6 %FI
450 1083 5 END;
451 1084 5
452 1085 4 END;
453 1086 4

```

```
454 1087 4      [(.major EQL maj_runoff) AND (.minor EQL min_page)] :
455 1088 4      |
456 1089 4      | Process a page number.
457 1090 4      |
458 1091 4      | process_page ();
459 1092 4      |
460 1093 4      [(.major EQL maj_runoff) AND (.minor EQL min_index)] :
461 1094 5      BEGIN
462 1095 5      |
463 1096 5      | Make an entry in the table of contents for the index
464 1097 5      |
465 1098 5      | errcnt = 0;                      ! Reset count of CONFUSED messages.
466 1099 5      |
467 1100 5      | IF .cmdblk [contents$v_tms11]
468 1101 5      | THEN
469 1102 6      | BEGIN
470 1103 6      | put ('');
471 1104 6      |
472 1105 6      | IF .hl_n NEQ 0 THEN put ('[va15]');
473 1106 6      |
474 1107 6      | put ('*c0n*Index@@');
475 1108 6      | END
476 1109 5      | ELSE
477 1110 6      | BEGIN
478 1111 6      | IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');
479 1112 6      |
480 1113 6      | IF .indents [0] NEQ 0
481 1114 6      | THEN
482 1115 6      | put ($str_concat ('.I-', $str_ascii (.indents [0])));
483 1116 6      |
484 1117 6      | put ('INDEX');
485 1118 6      | put ('.BLANK');
486 1119 5      | END;
487 1120 5      |
488 1121 5      | hl_n = 0;                      ! Set .BLANK info
489 1122 5      | lasthl = 0;
490 1123 4      | END;
491 1124 4      |
492 1125 5      [(.major EQL maj_runoff)
493 1126 5      AND (
494 1127 6      | (.minor EQL min_examp_inf)
495 1128 6      | OR (.minor EQL min_table_inf)
496 1129 6      | OR (.minor EQL min_figur_inf)
497 1130 5      | )
498 1131 4      ] :
499 1132 4      |
500 1133 4      | Example, Table, or Figure information
501 1134 4      |
502 1135 4      | process_entity_info ();
503 1136 4      |
504 1137 5      [(.major EQL maj_runoff)
505 1138 5      AND (
506 1139 6      | (.minor EQL min_examp_txt)
507 1140 6      | OR (.minor EQL min_examp_txt_u)
508 1141 6      | OR (.minor EQL min_figur_txt)
509 1142 6      | OR (.minor EQL min_figur_txt_u)
510 1143 6      | OR (.minor EQL min_table_txt)
```

511	1144	6	OR (.minor EQL min_table_txt_u)
512	1145	5)
513	1146	4] :
514	1147	4	Figure, Table or Example text
515	1148	4	process_entity_txt ();
516	1149	4	
517	1150	4	[(.major EQL maj_runoff) AND (.minor EQL min_hl_inf)] :
518	1151	4	BEGIN
519	1152	4	Read header level counters and display descriptors
520	1153	5	errcnt = 0; ! Reset count of CONFUSED messages.
521	1154	5	First get count of header level counters.
522	1155	5	\$xpo_get (iob = tociob, fullwords = 1);
523	1156	5	hlcnt = .(.toclob [iob\$a_data]);
524	1157	5	Next, get the 'n' value for this '.HL n' command
525	1158	5	\$xpo_get (iob = tociob, fullwords = 1);
526	1159	5	hl_n = .(.toclob [iob\$a_data]);
527	1160	5	Now read the complete set of header level counters,
528	1161	5	not all of which are important
529	1162	5	\$xpo_get (iob = tociob, fullwords = .hlcnt);
530	1163	5	BEGIN
531	1164	5	BIND
532	1165	5	x = .toclob [iob\$a_data] : VECTOR;
533	1166	5	INCR i FROM 1 TO .hlcnt DO
534	1167	5	BEGIN
535	1168	5	Copy the header level counters to a safe place
536	1169	5	hlhist [.i - 1] = .x [.i - 1];
537	1170	5	END;
538	1171	5	END;
539	1172	5	Finally, read the header level display descriptors.
540	1173	6	\$xpo_get (iob = tociob, fullwords = .hlcnt);
541	1174	6	BEGIN
542	1175	6	BIND
543	1176	6	x = .toclob [iob\$a_data] : VECTOR;
544	1177	6	INCR i FROM 1 TO .hlcnt DO
545	1178	6	BEGIN
546	1179	7	Copy the header level descriptors to a safe place
547	1180	7	
548	1181	7	
549	1182	7	
550	1183	7	
551	1184	6	
552	1185	6	
553	1186	5	
554	1187	5	
555	1188	5	
556	1189	5	
557	1190	5	
558	1191	6	
559	1192	6	
560	1193	6	
561	1194	6	
562	1195	6	
563	1196	6	
564	1197	7	
565	1198	7	
566	1199	7	
567	1200	7	

```

: 568      1201 7      hldsp [.i - 1] = .x [.i - 1];
569      1202 6      END;
: 570      1203 6
571      1204 5      END;
572      1205 4      END;
573      1206 4
574      1207 5      [(.major EQL maj_runoff)
575      1208 5      AND (
576      1209 6      (.minor EQL min_hl_txt)
577      1210 6      OR (.minor EQL min_hl_txt_u)
578      1211 5      )
579      1212 4      ] :
580      1213 5      BEGIN
581      1214 5      | Process the text part of a header level.
582      1215 5
583      1216 5      LOCAL
584      1217 5      needed_fill;
585      1218 5
586      1219 5      errcnt = 0;          ! Reset count of CONFUSED messages.
587      1220 5
588      1221 5      | First read the number of characters needed to
589      1222 5      | represent the header level
590      1223 5
591      1224 5      $xpo_get (iob = tociob, fullwords = 1);
592      1225 5      txtlen = .(.toclob [iob$a_data]); ! Save text length.
593      1226 5
594      1227 5
595      1228 5      | Generate a table of contents with or without text
596      1229 5      | The header level may or may not have text associated with it
597      1230 5      | Get the text, if there is any
598      1231 5
599      1232 5      IF .txtlen NEQ 0
600      1233 5      THEN
601      1234 5          $xpo_get (iob = tociob, fullwords = CH$ALLOCATION (.txtlen));
602      1235 5
603      1236 5
604      1237 5      | Generate the associated TOC text only if this header
605      1238 5      | level is within the depth specified by the user.
606      1239 5
607      1240 5      IF .hl_n LEQ .cmdblk [contents$g_headers]
608      1241 5      THEN
609      1242 6          BEGIN
610      1243 6
611      1244 7          IF (.hl_n LEQ .cmdblk [contents$g_double_space])
612      1245 6          AND NOT .cmdblk [contents$v_tms11]
613      1246 6          THEN
614      1247 6              put ('.BLANK');
615      1248 6
616      1249 6
617      1250 6          | Initialize output line.
618      1251 6
619      1252 6          clr_line ();
620      1253 6
621      1254 6          | All header level entries are generated under the
622      1255 6          | assumption of a .LMn command in the file.
623      1256 6          | Therefore, even though no text has been generated
624      1257 6          | yet we have to act as if 'n' characters have been
```

```

: 625
: 626
: 627
: 628
: 629
: 630
: 631
: 632
: 633
: 634
: 635
: 636
: 637
: 638
: 639
: 640
: 641
: 642
: 643
: 644
: 645
: 646
: 647
: 648
: 649
: 650
: 651
: 652
: 653
: 654
: 655
: 656
: 657
: 658
: 659
: 660
: 661
: 662
: 663
: 664
: 665
: 666
: 667
: 668
: 669
: 670
: 671
: 672
: 673
: 674
: 675
: 676
: 677
: 678
: 679
: 680
: 681

```

```

1258 6
1259 6
1260 6
1261 6
1262 6
1263 6
1264 6
1265 6
1266 6
1267 6
1268 6
1269 6
1270 6
1271 6
1272 6
1273 6
1274 6
1275 6
1276 6
1277 6
1278 6
1279 6
1280 6
1281 6
1282 6
1283 7
1284 7
1285 7
1286 7
1287 7
1288 7
1289 7
1290 6
1291 7
1292 7
1293 7
1294 7
1295 7
1296 6
1297 6
1298 6
1299 6
1300 6
1301 6
1302 6
1303 6
1304 6
1305 6
1306 6
1307 6
1308 6
1309 6
1310 6
1311 6
1312 6
1313 6
1314 6

```

```

: taken up.
extlin = .indents [0];
: Set standard indentation in front of this header level line
line_indent = .indents [0];
: Generate the header level numbers if the header is
: numbered. If unnumbered, just put out the text.
outsec ();
: Set up line-wrap point
wrap = .extlin;
: Generate the page number so as to get its length.
: This is needed so that the proper number of dots
: can be supplied.
ptrpag = CH$PTR (txtpag);
IF NOT .cmdblk [contents$v_standard_page]
THEN
BEGIN
: Use the running page counter instead of a
: real page number.
lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10);
END
ELSE
BEGIN
: Supply the usual page number.
lenpag = pacpag (pagen, ptrpag);
END;
: Set limit beyond which words must be wrapped
rmargin = .cmdblk [contents$g_page_width] - .lenpag - 1;
: Scan text and put it onto the line.
fmttxt (.txtlen, CH$PTR (.tociob [iob$a_data]));
: Insert page reference into line if desired.
insref ();
: Now actually put this line of text into the file
put ((.intlin, CH$PTR (line)));
clr_line ();

```

682	1315	5	END;
683	1316	5	
684	1317	5	lasthl = .hl_n;
685	1318	4	END;
686	1319	4	
687	1320	5	[(.major EQL maj_runoff)
688	1321	5	AND (
689	1322	6	(.minor EQL min_chapt_inf)
690	1323	6	OR (.minor EQL min_append_inf)
691	1324	5)
692	1325	4] :
693	1326	5	BEGIN
694	1327	5	! Process Chapter/Appendix info record
695	1328	5	!
696	1329	5	LOCAL
697	1330	5	len;
698	1331	5	
699	1332	5	
700	1333	5	\$xpo_get (iob = tociob, fullwords = 1); ! Get TOC page
701	1334	5	toc_page = ..toc_iob [iob\$a_data];
702	1335	5	\$xpo_get (iob = tociob, fullwords = 1); ! Skip unused word
703	1336	5	\$xpc_get (iob = tociob, fullwords = 1); ! Get prefix length
704	1337	5	len = ..toc_iob [iob\$a_data];
705	1338	5	
706	1339	5	IF .len GTR 0
707	1340	5	THEN
708	1341	5	\$xpo_get (iob = tociob, fullwords = CH\$ALLOCATION (.len)); ! Get prefix string
709	1342	5	
710	1343	5	\$str_copy (string = (.len, CH\$PTR (.toc_iob [iob\$a_data])),
711	1344	5	target = prestr);
712	1345	5	
713	1346	5	\$xpo_get (iob = tociob, fullwords = 1); ! Get suffix length
714	1347	5	len = ..toc_iob [iob\$a_data];
715	1348	5	
716	1349	5	IF .len GTR 0
717	1350	5	THEN
718	1351	5	\$xpo_get (iob = tociob, fullwords = CH\$ALLOCATION (.len)); ! Get suffix string
719	1352	5	
720	1353	5	\$str_copy (string = (.len, CH\$PTR (.toc_iob [iob\$a_data])),
721	1354	5	target = sfxstr);
722	1355	5	
723	1356	4	END;
724	1357	4	
725	1358	5	[(.major EQL maj_runoff)
726	1359	5	AND (
727	1360	6	(.minor EQL min_chapt)
728	1361	6	OR (.minor EQL min_chapt_txt)
729	1362	6	OR (.minor EQL min_chapt_txt_u)
730	1363	6	OR (.minor EQL min_append)
731	1364	6	OR (.minor EQL min_append_txt)
732	1365	6	OR (.minor EQL min_append_txt_u)
733	1366	5)
734	1367	4] :
735	1368	5	BEGIN
736	1369	5	!
737	1370	5	! Process the start of a chapter or appendix
738	1371	5	!

739	1372	5	LOCAL
740	1373	5	ptr;
741	1374	5	
742	1375	5	errcnt = 0; ! Reset count of CONFUSED messages.
743	1376	5	clr_line (); ! Clear the text buffer
744	1377	5	
745	1378	5	IF .cndblk [contents\$y_tms11]
746	1379	5	THEN
747	1380	6	BEGIN
748	1381	6	! TMS output
749	1382	6	! put ('');
750	1383	6	
751	1384	6	put ('');
752	1385	6	
753	1386	6	IF .hl_n NEQ 0 THEN put ('[va15]');
754	1387	6	
755	1388	6	IF NOT .toc_page
756	1389	6	THEN
757	1390	7	literal_text ('*c0n*')
758	1391	6	ELSE
759	1392	6	literal_text ('*c0np*');
760	1393	6	
761	1394	6	END
762	1395	5	ELSE
763	1396	6	BEGIN
764	1397	6	! RUNOFF output
765	1398	6	! put ('.TEST PAGE 5');
766	1399	6	
767	1400	6	put ('.TEST PAGE 5');
768	1401	6	
769	1402	6	IF .cndblk [contents\$g_headers] NEQ 0
770	1403	6	THEN
771	1404	7	BEGIN
772	1405	7	
773	1406	7	IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');
774	1407	7	
775	1408	6	END;
776	1409	6	
777	1410	6	IF .indents [0] NEQ 0
778	1411	6	THEN
779	1412	6	put (\$str_concat ('.I-', \$str_ascii (.indents [0])));
780	1413	6	
781	1414	6	line_indent = .indents [0];
782	1415	6	rmargin = .cndblk [contents\$g_page_width];
783	1416	5	END;
784	1417	5	
785	1418	5	! Now reset the .BLANK information
786	1419	5	! lasthl = 0;
787	1420	5	! hl_n = 0;
788	1421	5	
789	1422	5	
790	1423	5	
791	1424	6	IF (.minor NEQ min_chapt_txt_u)
792	1425	6	AND (.minor NEQ min_append_txu)
793	1426	5	THEN
794	1427	6	BEGIN
795	1428	6	!


```
796 1429 6      ! Insert the word CHAPTER or APPENDIX
797 1430 6
798 1431 7      IF (.minor EQL min_chapt)
799 1432 7      OR (.minor EQL min_chapt_txt)
800 1433 6      THEN
801 1434 7          BEGIN
802 1435 7              ! This is the start of a chapter
803 1436 7              IF .cmdblk [contents$y_tms11]
804 1437 7              THEN
805 1438 7                  fmttxt (8, CH$PTR (UPLIT ('Chapter ')))
806 1439 7              ELSE
807 1440 7                  fmttxt (8, CH$PTR (UPLIT ('CHAPTER ')));
808 1441 7
809 1442 7          END
810 1443 7      ELSE
811 1444 7          BEGIN
812 1445 6              ! This is the start of an appendix
813 1446 7              IF .cmdblk [contents$y_tms11]
814 1447 7              THEN
815 1448 7                  fmttxt (9, CH$PTR (UPLIT ('Appendix ')))
816 1449 7              ELSE
817 1450 7                  fmttxt (9, CH$PTR (UPLIT ('APPENDIX ')));
818 1451 7
819 1452 7          END;
820 1453 6
821 1454 6      ! Insert the prefix string if any
822 1455 6      IF .prestr [str$h_length] NEQ 0
823 1456 6      THEN
824 1457 6          fmttxt (.prestr [str$h_length], .prestr [str$a_pointer]);
825 1458 6
826 1459 6      ! Convert the 'number' of this chapter/appendix to its visual form.
827 1460 6      ptrsct = CH$PTR (txtsct);
828 1461 6      lensct = pacsec (pagen, ptrsct);
829 1462 6      fmttxt (.lensct, CH$PTR (txtsct));
830 1463 6
831 1464 6      ! Insert the suffix string if there is one
832 1465 6      IF .sfxstr [str$h_length] NEQ 0
833 1466 6      THEN
834 1467 6          fmttxt (.sfxstr [str$h_length], .sfxstr [str$a_pointer]);
835 1468 6
836 1469 6      IF NOT .cmdblk [contents$y_tms11] THEN write_char (%C' ', counts_visually);
837 1470 6      END;
838 1471 5      IF .cmdblk [contents$y_tms11] THEN write_char (%C'a');
839 1472 5
840 1473 5      ! Determine the number of characters in the chapter/appendix title
841 1474 5
842 1475 5
843 1476 5
844 1477 5
845 1478 5
846 1479 5
847 1480 5
848 1481 5
849 1482 5
850 1483 5
851 1484 5
852 1485 5
```

853	1486	5	if any.
854	1487	5	
855	1488	5	\$xpo_get (iob = tociob, fullwords = 1);
856	1489	5	txtlen = (.toclob [iob\$a_data]); !Save text length.
857	1490	5	
858	1491	5	Now read the chapter/appendix text if
859	1492	5	any exists.
860	1493	5	
861	1494	5	IF .txtlen GTR 0
862	1495	5	THEN
863	1496	5	\$xpo_get (iob = tociob, fullwords = CH\$ALLOCATION (.txtlen));
864	1497	5	
865	1498	5	IF NOT .cmdblk [contents\$y_tms11]
866	1499	5	THEN
867	1500	5	pad ((.indents [0] + .indents [1] - .extlin));
868	1501	5	
869	1502	5	
870	1503	5	Be sure that long chapter titles get wrapped around correctly
871	1504	5	
872	1505	5	wrap = .extlin;
873	1506	5	
874	1507	5	IF .toc_page
875	1508	5	THEN
876	1509	6	BEGIN
877	1510	6	
878	1511	6	Generate a page number.
879	1512	6	
880	1513	6	ptrpag = CH\$PTR (txtpag);
881	1514	6	
882	1515	6	IF NOT .cmdblk [contents\$y_standard_page]
883	1516	6	THEN
884	1517	6	lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10)
885	1518	6	ELSE
886	1519	6	lenpag = pacpag (pagen, ptrpag);
887	1520	6	
888	1521	6	rmargin = .cmdblk [contents\$g_page_width] - .lenpag - 1;
889	1522	5	END;
890	1523	5	
891	1524	5	
892	1525	5	Scan text and put it onto the line.
893	1526	5	
894	1527	5	fmttxt (.txtlen, CH\$PTR (.toclob [iob\$a_data]));
895	1528	5	
896	1529	5	IF .toc_page
897	1530	5	THEN
898	1531	6	BEGIN
899	1532	6	
900	1533	6	Insert page reference into line
901	1534	6	
902	1535	6	insref ();
903	1536	6	toc_page = false;
904	1537	6	END
905	1538	5	ELSE
906	1539	6	BEGIN
907	1540	6	
908	1541	6	IF .cmdblk [contents\$y_tms11] THEN write_char (%C'a');
909	1542	5	END;

```

910 1543 5
911 1544 5
912 1545 5
913 1546 5
914 1547 5
915 1548 5
916 1549 5
917 1550 5
918 1551 5
919 1552 5
920 1553 4
921 1554 4
922 1555 4
923 1556 5
924 1557 5
925 1558 5
926 1559 5
927 1560 5
928 1561 5
929 1562 5
930 1563 5
931 1564 5
932 1565 5
933 1566 5
934 1567 5
935 1568 5
936 1569 5
937 1570 5
938 1571 5
939 1572 5
940 1573 5
941 1574 5
942 1575 5
943 1576 5
944 1577 5
945 1578 5
946 1579 6
947 1580 6
948 1581 6
949 1582 6
950 1583 6
951 1584 6
952 1585 6
953 1586 6
954 1587 7
955 1588 7
956 1589 7
957 1590 7
958 1591 7
959 1592 7
960 1593 6
961 1594 7
962 1595 7
963 1596 7
964 1597 7
965 1598 7
966 1599 7

      Now actually put this line of text into the file
      put ((.intlin, CH$PTR (line)));
      clr_line ();

      Standard skipping after starting chapter/appendix entry
      IF NOT .cmdblk [contents$v_tms11] THEN put ('.BLANK');
      END;

[.major EQL maj_send_contents] :
BEGIN
      A SEND CONTENTS record
      LOCAL
      type,
      position,
      len,
      iob_ptr : REF $xpo_iob ();

      type = .minor<0, %BPVAL/2>;
      position = .minor<%BPVAL/2, %BPVAL/2>;
      $xpo_get (iob = tociob, fullwords = 1);      ! Get text length
      len = ..toc_iob [iob$a_data];

      IF .len GTR 0
      THEN
          $xpo_get (iob = tociob, fullwords = CH$ALLOCATION (.len));      ! Get text

      SELECT ONE .position OF
      SET
      [scon lev]:
      BEGIN
          Insert in main part of table of contents
          iob_ptr = 0;

          IF .type EQL scon_comm
          THEN
              BEGIN
                  A command. Insert directly into .RNT file
                  put ((.len, CH$PTR (.toc_iob [iob$a_data])));
              END
          ELSE
              BEGIN
                  Text to be inserted into the RNT
                  hl_n = 1;      ! Treat as HL 1
              END
          END
      END
  
```

967 1600 7
968 1601 7
969 1602 7
970 1603 7
971 1604 7
972 1605 7
973 1606 7
974 1607 7
975 1608 7
976 1609 7
977 1610 7
978 1611 7
979 1612 7
980 1613 7
981 1614 7
982 1615 7
983 1616 7
984 1617 7
985 1618 7
986 1619 7
987 1620 8
988 1621 8
989 1622 8
990 1623 8
991 1624 8
992 1625 8
993 1626 8
994 1627 8
995 1628 6
996 1629 8
997 1630 8
998 1631 8
999 1632 8
1000 1633 7
1001 1634 7
1002 1635 7
1003 1636 7
1004 1637 7
1005 1638 7
1006 1639 7
1007 1640 7
1008 1641 8
1009 1642 8
1010 1643 8
1011 1644 8
1012 1645 7
1013 1646 8
1014 1647 8
1015 1648 8
1016 1649 7
1017 1650 7
1018 1651 7
1019 1652 7
1020 1653 7
1021 1654 6
1022 1655 6
1023 1656 5

```
IF .type EQL scon_txt THEN toc_page = true;
Set minor code to unnumbered HL so OUTSEC
won't generate a section number.
minor = min_hl_txt_u;
Initialize formatting parameters
clr_line ();
extlin = .indents [0];
line_indent = .indents [0];
Call OUTSEC to indent line to proper point
outsec ();
wrap = .extlin;
IF .toc_page
THEN
BEGIN
Generate page number to get its length
ptrpag = CH$PTR (txtpag);
IF NOT .cmdblk [contents$v_standard_page]
THEN
lenpag = pacbas (.pagen [sct_run_pag^], ptrpag, 10)
ELSE
lenpag = pacpag (pagen, ptrpag);
END
ELSE
lenpag = 0;
rmargin = .cmdblk [contents$g_page_width] - .lenpag - 1;
fmttxt (.len, CH$PTR (.tociob [iob$a_data]));
IF .toc_page
THEN
BEGIN
insref ();
toc_page = false;
END
ELSE
BEGIN
IF .cmdblk [contents$v_tms11] THEN write_char (%C'a');
END;
put ((.intl, CH$PTR (line)));
clr_line ();
lasthl = .hl_n;
END;
END;
```

```
1024 1657 5
1025 1658 5      [scon_exa]:          ! Insert with examples
1026 1659 5      iob_ptr = exiob;
1027 1660 5
1028 1661 5      [scon_fig]:          ! Insert with figures
1029 1662 5      iob_ptr = fgiob;
1030 1663 5
1031 1664 5      [scon_tab]:          ! Insert with tables
1032 1665 5      iob_ptr = tbiob;
1033 1666 5
1034 1667 5      TES;
1035 1668 5
1036 1669 6      IF (.iob_ptr NEQ 0)
1037 1670 5      AND vfyca (new_captions)
1038 1671 5      THEN
1039 1672 6          BEGIN
1040 1673 6              SEND CONTENTS is to be inserted with FIGURES, TABLES
1041 1674 6              or EXAMPLES. Write record to appropriate temp file.
1042 1675 6
1043 1676 6          LOCAL
1044 1677 6              record_header : VECTOR [3];
1045 1678 6
1046 1679 6              record_header [0] = .major;
1047 1680 6              record_header [1] = .minor;
1048 1681 6              record_header [2] = .len;
1049 1682 6              $xpo_put (iob = .iob_ptr, binary_data = (3, record_header));
1050 1683 6
1051 1684 6              IF .len NEQ 0
1052 1685 6              THEN
1053 1686 6                  $xpo_put (iob = .iob_ptr,
1054 1687 6                      binary_data = (CR$ALLOCATION (.len), .tociob [iobsa_data]));
1055 1688 6
1056 1689 6
1057 1690 5          END;
1058 1691 5
1059 1692 4      END;
1060 1693 4
1061 1694 4      [.major EQL maj_send] :
1062 1695 5      BEGIN
1063 1696 5          Process text given on a .SEND command
1064 1697 5
1065 1698 5          NOTE: When the user said .SEND TOC n,text the value of 'n'
1066 1699 5          was written to the file too. It is now available as
1067 1700 5          the value saved in minor. If the user didn't
1068 1701 5          specify n, then 0 was passed.
1069 1702 5
1070 1703 5          NOTE: The value of 'n' has the following meanings:
1071 1704 5
1072 1705 5              0      insert directly into .RNT file
1073 1706 5
1074 1707 5              1      Example caption
1075 1708 5              -1     insert directly into .RNT file with
1076 1709 5              example captions.
1077 1710 5
1078 1711 5              2      Figure caption
1079 1712 5              -2     insert directly into .RNT file with
1080 1713 5
```

1081	1714	5	example captions.
1082	1715	5	
1083	1716	5	3 Table caption
1084	1717	5	-3 insert directly into .RNT file with
1085	1718	5	table captions
1086	1719	5	
1087	1720	5	n>3 Caption with no title or lead-in
1088	1721	5	-n insert directly into .RNT file with
1089	1722	5	+n captions
1090	1723	5	
1091	1724	5	Captions are parsed for a leading word followed by a
1092	1725	5	comma. This word is used for the figure number.
1093	1726	5	Captions are treated like header levels in that they
1094	1727	5	are followed by dots and the chapter-page number.
1095	1728	5	
1096	1729	5	LOCAL
1097	1730	5	ptr;
1098	1731	5	
1099	1732	5	errcnt = 0; ! Reset count of CONFUSED messages.
1100	1733	5	
1101	1734	5	Determine the number of characters specified on the .SEND command
1102	1735	5	
1103	1736	5	\$xpo_get (iob = tociob, fullwords = 1);
1104	1737	5	txtlen = .(.toclob [iob\$a_data]); !Save text length.
1105	1738	5	
1106	1739	5	If there was text given process it.
1107	1740	5	
1108	1741	5	IF .txtlen GTR 0
1109	1742	5	THEN
1110	1743	6	BEGIN
1111	1744	6	There was text given.
1112	1745	6	Now actually read the text
1113	1746	6	
1114	1747	6	LOCAL
1115	1748	6	record_header : VECTOR [3];
1116	1749	6	
1117	1750	6	\$xpo_get (iob = tociob, fullwords = CH\$ALLOCATION (.txtlen));
1118	1751	6	record_header [0] = maj_send;
1119	1752	6	record_header [1] = .minor;
1120	1753	6	record_header [2] = .txtlen;
1121	1754	6	
1122	1755	6	IF .minor EQL 0
1123	1756	6	THEN
1124	1757	6	BEGIN
1125	1758	7	Insert text directly into .RNT file
1126	1759	7	
1127	1760	7	insrnt (.txtlen, CH\$PTR (.toclob [iob\$a_data]));
1128	1761	7	
1129	1762	7	END
1130	1763	7	ELSE
1131	1764	6	BEGIN
1132	1765	7	Miscellaneous caption or send
1133	1766	7	
1134	1767	7	IF (ABS (.minor) GTR 3) OR ((ABS (.minor) LEQ 3) AND vfyap (old_captions))
1135	1768	7	THEN
1136	1769	8	
1137	1770	7	

```

: 1138      1771 8      BEGIN
: 1139      1772 9
: 1140      1773 8
: 1141      1774 8      | Not a figure, table or example caption
: 1142      1775 8      | OR a figure, table or example caption
: 1143      1776 8      | and we're using old style (SEND TOC) captions
: 1144      1777 8      |
: 1145      1778 8      | nextstc = MIN (ABS (.minor), .nextstc);
: 1146      P 1779 8      | $xpo_put (iob = stciob, binary_data = (3, record_header));
: 1147      1780 8      | $xpo_put (iob = stciob,
: 1148      1781 7      |     binary_data = (CH$ALLOCATION (.txtlen), .tociob [iob$a_data]));
: 1149      1782 7      | END;
: 1150      1783 6      |
: 1151      1784 6      |
: 1152      1785 5      |
: 1153      1786 5      |
: 1154      1787 4      |
: 1155      1788 4      |
: 1156      1789 4      | [OTHERWISE] :
: 1157      1790 5      | BEGIN
: 1158      1791 5      |     errcnt = .errcnt + 1;
: 1159      1792 5      |
: 1160      1793 5      |     IF .errcnt LEQ 5
: 1161      1794 5      |     THEN
: 1162      L 1795 5      |     %IF %BLISS (BLISS32)
: 1163      1796 5      |     %THEN
: 1164      1797 5      |         SIGNAL (contents$_invrecord, 1, tociob [iob$t_resultant]);
: 1165      U 1798 5      |     %ELSE
: 1166      U 1799 5      |         $xpo_put_msg (severity = warning,
: 1167      U 1800 5      |             string = $str_concat ('invalid record group in file ',
: 1168      U 1801 5      |                 tociob [iob$t_resultant], ''));
: 1169      1802 5      |     %FI
: 1170      1803 5      |
: 1171      1804 4      |
: 1172      1805 4      |     END;
: 1173      1806 4      |
: 1174      1807 3      |
: 1175      1808 3      |
: 1176      1809 2      |
: 1177      1810 2      |
: 1178      1811 2      |     Finished processing file.
: 1179      1812 2      |
: 1180      1813 2      |
: 1181      1814 2      |
: 1182      1815 2      |     IF .cmdblk [contents$v_log]
: 1183      1816 2      |     THEN
: 1184      L 1817 2      |     %IF %BLISS (BLISS32)
: 1185      1818 2      |     %THEN
: 1186      1819 2      |         SIGNAL (contents$_complete, 1, tociob [iob$t_resultant]);
: 1187      U 1820 2      |     %ELSE
: 1188      U 1821 2      |         $xpo_put_msg (severity = success,
: 1189      U 1822 2      |             string = $str_concat ('processing complete ', tociob [iob$t_resultant], ''));
: 1190      1823 2      |     %FI
: 1191      1824 2      |
: 1192      1825 2      |     $xpo_close (iob = tociob);
: 1193      1826 2      |     RETURN true;
: 1194      1827 1      |     END;
:                |
:                |     Close input file
:                |     Successful completion (by definition)
:                |     End of toc

```

```

;
.TITLE CONTENTS CONTENTS - Table of Contents formatter
       program
.IDENT \V04-000\
.PSECT $SPLITS,NOWRT,NOEXE,2

00000 P.AAA: .BLKB 0
40 40 78 65 64 6E 5D 35 31 61 76 5B 00000 P.AAB: .ASCII \[va15]\
49 2A 6E 30 63 2A 00000 P.AAC: .ASCII \*c0n*Indexaa\
32 4B 53 2E 00012 P.AAD: .ASCII \.SK2\
31 4B 53 2E 00016 P.AAE: .ASCII \.SK1\
2D 49 2E 0001A P.AAG: .ASCII \.I-\
58 45 44 4E 49 0001D P.AAI: .ASCII \INDEX\
4B 4E 41 4C 42 2E 00022 P.AAJ: .ASCII \.BLANK\
4B 4E 41 4C 42 2E 00028 P.AAK: .ASCII \.BLANK\
0002E P.AAL: .BLKB 0
5D 35 31 61 76 5B 0002E P.AAM: .ASCII \[va15]\
00 00 00 2A 6E 30 63 2A 00034 P.AAN: .ASCII \*c0n*\<0><0><0>
00 00 2A 70 6E 30 63 2A 0003C P.AAO: .ASCII \*c0np*\<0><0>
35 20 45 47 41 50 20 54 53 45 54 2F 00044 P.AAP: .ASCII \.TEST PAGE 5\
32 4B 53 2E 00050 P.AAQ: .ASCII \.SK2\
31 4B 53 2E 00054 P.AAR: .ASCII \.SK1\
2D 49 2E 00058 P.AAT: .ASCII \.I-\
0005B .BLKB 1
00 00 00 20 72 65 74 70 61 68 43 0005C P.AAV: .ASCII \Chapter \
00 00 00 20 78 52 45 54 50 41 48 43 00064 P.AAW: .ASCII \CHAPTER \
00 00 00 20 58 69 64 6E 65 70 70 41 0006C P.AAX: .ASCII \Appendix \<0><0><0>
00 00 00 20 58 49 44 4E 45 50 50 41 00078 P.AAY: .ASCII \APPENDIX \<0><0><0>
4B 4E 41 4C 42 2E 00084 P.AAZ: .ASCII \.BLANK\

.PSECT $OWNS,NOEXE,2

00000 TOCJOB: .BLKB 244
000F4 FILE_TYPE:
          .BLKB 4
000F8 FGI0B: .BLKB 244
001EC TBI0B: .BLKB 244
002E0 EXI0B: .BLKB 244
003D4 HLCNT: .BLKB 4
003D8 LASTHL: .BLKB 4
00000000 003DC TOC_PAGE:
          .LONG 0
003E0 CAPTYP: .BLKB 4
003E4 CAPMSG: .BLKB 4
003E8 ERRCNT: .BLKB 4
0000 003EC $STR$STRING:
          .WORD 0
01 0E 003EE .BYTE 14, 1
00000000 003F0 .ADDRESS P.AAA
0006 003F4 $STR$STRING:
          .WORD 6
01 0E 003F6 .BYTE 14, 1
00000000 003F8 .ADDRESS P.AAB
000C 003FC $STR$STRING:
          .WORD 12

```



```

01 0E 003FE .BYTE 14, 1
00000000' 00400 .ADDRESS P.AAC
0004 00404 $STR$STRING:
                                .WORD 4
01 0E 00406 .BYTE 14, 1
00000000' 00408 .ADDRESS P.AAD
0004 0040C $STR$STRING:
                                .WORD 4
01 0E 0040E .BYTE 14, 1
00000000' 00410 .ADDRESS P.AAE
0003 00414 $STR$STRING:
                                .WORD 3
01 0E 00416 .BYTE 14, 1
00000000' 00418 .ADDRESS P.AAG
0005 0041C $STR$STRING:
                                .WORD 5
01 0E 0041E .BYTE 14, 1
00000000' 00420 .ADDRESS P.AAI
0006 00424 $STR$STRING:
                                .WORD 6
01 0E 00426 .BYTE 14, 1
00000000' 00428 .ADDRESS P.AAJ
0006 0042C $STR$STRING:
                                .WORD 6
01 0E 0042E .BYTE 14, 1
00000000' 00430 .ADDRESS P.AAK
0000 00434 $STR$STRING:
                                .WORD 0
01 0E 00436 .BYTE 14, 1
00000000' 00438 .ADDRESS P.AAL
0006 0043C $STR$STRING:
                                .WORD 6
01 0E 0043E .BYTE 14, 1
00000000' 00440 .ADDRESS P.AAM
000C 00444 $STR$STRING:
                                .WORD 12
01 0E 00446 .BYTE 14, 1
00000000' 00448 .ADDRESS P.AAP
0004 0044C $STR$STRING:
                                .WORD 4
01 0E 0044E .BYTE 14, 1
00000000' 00450 .ADDRESS P.AAQ
0004 00454 $STR$STRING:
                                .WORD 4
01 0E 00456 .BYTE 14, 1
00000000' 00458 .ADDRESS P.AAR
0003 0045C $STR$STRING:
                                .WORD 3
01 0E 0045E .BYTE 14, 1
00000000' 00460 .ADDRESS P.AAT
0006 00464 $STR$STRING:
                                .WORD 6
01 0E 00466 .BYTE 14, 1
00000000' 00468 .ADDRESS P.AAZ

```

IOB\$= TOC IOB
IOB\$= TOC IOB

```
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
I0B$= TOC I08  
  
.EXTRN DSRTOC$_BADVALUE  
.EXTRN DSRTOC$_OPENIN, DSRTOC$_OPENOUT  
-XTRN DSRTOC$_VALERR, DSRTOC$_CAPTIONS  
  XTRN DSRTOC$_CLOSEQUOT  
  XTRN DSRTOC$_CONFQUAL  
.EXTRN DSRTOC$_CTRLCHAR  
.EXTRN DSRTOC$_EMPTYIN  
.EXTRN DSRTOC$_IGNORED  
.EXTRN DSRTOC$_INVINPUT  
.EXTRN DSRTOC$_INVRECORD  
.EXTRN DSRTOC$_OVERSTRK  
.EXTRN DSRTOC$_COMPLETE  
.EXTRN DSRTOC$_CREATED  
.EXTRN DSRTOC$_IGNORENEW  
.EXTRN DSRTOC$_IGNOREOLD  
.EXTRN DSRTOC$_PROCFILE  
.EXTRN DSRTOC$_TEXTD, DSRTOC$__TMS11  
.EXTRN DSRTOC$_IDENT, CMDBLK  
.EXTRN TOCOOB, STCIOB, CNTVRL  
.EXTRN CNTVRP, CHRUT, TMPSTR  
.EXTRN TXTLEN, HL N, HLLIST  
.EXTRN HLDSP, MAJOR, MINOR  
.EXTRN PAGEN, NXTSTC, RNO EX  
.EXTRN RNO_FG, RNO_TB, TMS_EX  
.EXTRN TMS-FG, TMS-TB, PRESTR  
.EXTRN SFXSTR, LP, INTLIN  
.EXTRN EXT LIN, LINE, LENPAG  
.EXTRN TXTPAG, PTRPAG, LENSCT  
.EXTRN TXTSCT, PTRSCT, RMARGIN  
.EXTRN WRAP, LINE INDENT  
.EXTRN FMTTXT, INSREF, HDR_X  
.EXTRN STCHDR, INSRNT, SPLIT  
.EXTRN PACBAS, PACPAG, PACSEC  
.EXTRN PACXXX, OPEN ERROR  
.EXTRN XPOSGET, XPOSFAILURE  
.EXTRN XSTS COPY, STRSF AILURE  
.EXTRN XPOSPUT, XSTSJOIN
```

				00000000G	EF	9E	00002		.ENTRY	TOC, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	0974
				5A 00000000G	EF	9E	00009		MOVAB	XPOS\$FAILURE, R11	
				59 00000000G	EF	9E	00010		MOVAB	CMDBLK, R10	
				58 00000000G	EF	9E	00017		MOVAB	IOBS+68, R9	
				57 000000000	EF	9E	0001E		MOVAB	\$STR\$TARGET, R8	
				5E	14	C2	00025		MOVAB	IOBS+52, R7	
00000000V				EF	00	FB	00028		SUBL2	#20, SP	
				03	50	EB	0002F		CALLS	#0, TOCINI	1029
					10C6	31	00032		BLBS	R0, 1\$	
				67	08	B0	00035	1\$:	BRW	138\$	
	02			A7	02	90	00038		MOVW	#8, IOBS+52	1035
	F8			A7	06	90	0003C		MOVB	#2, IOBS+54	
					5B	DD	00040		MOVB	#6, IOBS+44	
					7E	D4	00042		PUSHL	R11	
		CC			A7	9F	00044		CLRL	-(SP)	
00000000G				EF	03	FB	00047		PUSHAB	IOBS	
00208001	8F				50	D1	0004E		CALLS	#3, XPOS\$GET	
					03	13	00055		CMPL	R0, #2129921	
					1075	31	00057		BEQL	2\$	
	52	04			A7	D0	0005A	2\$:	BRW	136\$	
					50	D4	0005E		MOVL	TOCIOB+56, R2	1039
FFFFFFFFF	8F	00C0			C7	D1	00060		CLRL	R0	1041
					24	12	00069		CMPL	FILE_TYPE, #-1	
					50	D6	0006B		BNEQ	3\$	
	02				62	D1	0006D		INCL	R0	
					1D	13	00070		CMPL	(R2), #2	
67	04	A2			04	A5	00072		BEQL	3\$	
	02	A7			02	90	00077		MULW3	#4, 4(R2), IOBS+52	1047
	F8	A7			06	90	0007B		MOVB	#2, IOBS+54	
					5B	DD	0007F		MOVB	#6, IOBS+44	
					7E	D4	00081		PUSHL	R11	
		CC			A7	9F	00083		CLRL	-(SP)	
00000000G	EF				03	FB	00086		PUSHAB	IOBS	
					A6	11	0008D		CALLS	#3, XPOS\$GET	
	19				50	E9	0008F	3\$:	BRB	1\$	1041
	67				08	B0	00092		BLBC	R0, 4\$	1054
	02	A7			02	90	00095		MOVW	#8, IOBS+52	
	F8	A7			06	90	00099		MOVB	#2, IOBS+54	
					5B	DD	0009D		MOVB	#6, IOBS+44	
					7E	D4	0009F		PUSHL	R11	
		CC			A7	9F	000A1		CLRL	-(SP)	
00000000G	EF				03	FB	000A4		PUSHAB	IOBS	
00000000G	EF				62	D0	000AB	4\$:	CALLS	#3, XPOS\$GET	
00000000G	EF	04			A2	D0	000B2		MOVL	(R2), MAJOR	1056
	54	00000000G			EF	D0	000BA		MOVL	4(R2), MINOR	1057
	01				54	D1	000C1		MOVL	MAJOR, R4	1062
					22	12	000C4		CMPL	R4, #1	
FFFFFFFFF	8F	00C0			C7	D1	000C6		BNEQ	6\$	
					09	12	000CF		CMPL	FILE_TYPE, #-1	1067
	03	00000000G			EF	D1	000D1		BNEQ	5\$	
					36	13	000D8		CMPL	MINOR, #3	1068
		E8			A7	9F	000DA	5\$:	BEQL	9\$	
									PUSHAB	TOCIOB+28	1077

		01	DD	000DD	PUSHL	#1		
		8F	DD	000DF	PUSHL	#DSRTOCS_INVINPUT		
		0FDD	31	000E5	BRW	134\$		
		52	D4	000E8	CLRL	R2		1087
02		54	D1	000EA	CMPL	R4, #2		
		02	12	000ED	BNEQ	7\$		
		52	D6	000EF	INCL	R2		
		50	D4	000F1	CLRL	R0		
01	00000000G	EF	D1	000F3	CMPL	MINOR, #1		
		02	12	000FA	BNEQ	8\$		
		50	D6	000FC	INCL	R0		
51		52	D2	000FE	MCOML	R2, R1		
50		51	CA	00101	BICL2	R1, R0		
01		50	D1	00104	CMPL	R0, #1		
		0A	12	00107	BNEQ	10\$		
00000000V	EF	00	FB	00109	CALLS	#0, PROCESS_PAGE		1091
		FF22	31	00110	BRW	1\$		
		50	D4	00113	CLRL	R0		1093
05	00000000G	EF	D1	00115	CMPL	MINOR, #5		
		02	12	0011C	BNEQ	11\$		
		50	D6	0011E	INCL	R0		
51		52	D2	00120	MCOML	R2, R1		
50		51	CA	00123	BICL2	R1, R0		
01		50	D1	00126	CMPL	R0, #1		
		03	13	00129	BEQL	12\$		
		01EF	31	0012B	BRW	24\$		
03		03B4	C7	D4	0012E	CLRL	ERRCNT	1098
	6A	01	E0	00132	BBS	#1, CMDBLK, 13\$		1100
		009B	31	00136	BRW	16\$		
		00000000G	EF	9F	00139	PUSHAB	STR\$FAILURE	1103
		7E	D4	0013F	CLRL	-(SP)		
		03B8	58	DD	00141	PUSHL	R8	
		7E	9F	00143	PUSHAB	\$STR\$STRING		
		7E	D4	00147	CLRL	-(SP)		
00000000G	EF	05	FB	00149	CALLS	#5, XST\$COPY		
00000000G	50	68	3C	00150	MOVZWL	TMPSTR, R0		
	69	50	C0	00153	ADDL2	R0, CHROUT		
E8	A9	68	9E	0015A	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
		07	90	0015D	MOVB	#7, IOB\$+44		
		58	DD	00161	PUSHL	R11		
		7E	D4	00163	CLRL	-(SP)		
		BC	A9	9F	00165	PUSHAB	IOB\$	
07	00000000G	EF	03	FB	00168	CALLS	#3, XPOS\$PUT	
		6A	01	E1	0016F	BBC	#1, CMDBLK, 14\$	
00000000G	EF	00	FB	00173	CALLS	#0, SPLIT		
		00000000G	EF	D5	0017A	TSTL	HL N	1105
		41	13	00180	BEQL	15\$		
		00000000G	EF	9F	00182	PUSHAB	STR\$FAILURE	
		7E	D4	00188	CLRL	-(SP)		
		03C0	58	DD	0018A	PUSHL	R8	
		7E	9F	0018C	PUSHAB	\$STR\$STRING		
		7E	D4	00190	CLRL	-(SP)		
00000000G	EF	05	FB	00192	CALLS	#5, XST\$COPY		
	50	68	3C	00199	MOVZWL	TMPSTR, R0		
00000000G	EF	50	C0	0019C	ADDL2	R0, CHROUT		
	69	68	9E	001A3	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
E8	A9	07	90	001A6	MOVB	#7, IOB\$+44		

07	00000000G	EF	BC	5B DD 001AA	PUSHL R11	
		6A		7E D4 001AC	CLRL -(SP)	
	00000000G	EF		A9 9F 001AE	PUSHAB IOB\$	
				03 FB 001B1	CALLS #3, XPOS\$PUT	
	00000000G			01 E1 001B8	BBC #1, CMDBLK, 15\$	
				00 FB 001BC	CALLS #0, SPLIT	
			00000000G	EF 9F 001C3	PUSHAB STR\$FAILURE	1107
				7E D4 001C9	CLRL -(SP)	
			03C8	5B DD 001CB	PUSHL R8	
				C7 9F 001CD	PUSHAB \$STR\$STRING	
			010A	31 001D1	BRW 22\$	
	00000000G			EF D5 001D4	TSTL HL,N	1111
				10 13 001DA	BEQL 17\$	
	00000000G			EF 9F 001DC	PUSHAB STR\$FAILURE	
				7E D4 001E2	CLRL -(SP)	
			03D0	5B DD 001E4	PUSHL R8	
				C7 9F 001E6	PUSHAB \$STR\$STRING	
				0E 11 001EA	BRB 18\$	
	00000000G			EF 9F 001EC	PUSHAB STR\$FAILURE	17\$:
				7E D4 001F2	CLRL -(SP)	
			03D8	5B DD 001F4	PUSHL R8	
				C7 9F 001F6	PUSHAB \$STR\$STRING	
	00000000G	EF		7E D4 001FA	CLRL -(SP)	
		50		05 FB 001FC	CALLS #5, XST\$COPY	
	00000000G	EF		68 3C 00203	MOVZWL TMPSTR, R0	
		69		50 C0 00206	ADDL2 R0, CHROUT	
	E8	A9		68 9E 0020D	MOVAB \$IOB\$OUTPUT, IOB\$+68	
				07 90 00210	MOVB #7, IOB\$+44	
				5B DD 00214	PUSHL R11	
			BC	7E D4 00216	CLRL -(SP)	
	00000000G	EF		A9 9F 00218	PUSHAB IOB\$	
		6A		03 FB 0021B	CALLS #3, XPOS\$PUT	
	00000000G	EF		01 E1 00222	BBC #1, CMDBLK, 19\$	
		50		00 FB 00226	CALLS #0, SPLIT	
			20	AA D0 0022D	MOVL INDENTS, R0	1113
				5C 13 00231	BEQL 20\$	
				7E D4 00233	CLRL -(SP)	1115
				50 DD 00235	PUSHL R0	
	00000000G	7E	0903	8F 3C 00237	MOVZWL #2307, -(SP)	
		EF		03 FB 0023C	CALLS #3, XST\$ASCII	
				50 DD 00243	PUSHL R0	
			03E0	C7 9F 00245	PUSHAB \$STR\$STRING0	
	00000000G	EF		02 FB 00249	CALLS #2, XST\$JOIN	
			00000000G	EF 9F 00250	PUSHAB STR\$FAILURE	
				7E D4 00256	CLRL -(SP)	
			0101	8F BB 00258	PUSHR #^M<R0,R8>	
	00000000G	EF		7E D4 0025C	CLRL -(SP)	
		50		05 FB 0025E	CALLS #5, XST\$COPY	
	00000000G	EF		68 3C 00265	MOVZWL TMPSTR, R0	
		69		50 C0 00268	ADDL2 R0, CHROUT	
	E8	A9		68 9E 0026F	MOVAB \$IOB\$OUTPUT, IOB\$+68	
				07 90 00272	MOVB #7, IOB\$+44	
				5B DD 00276	PUSHL R11	
				7E D4 00278	CLRL -(SP)	
			BC	A9 9F 0027A	PUSHAB IOB\$	
	00000000G	EF		03 FB 0027D	CALLS #3, XPOS\$PUT	
07		6A		01 E1 00284	BBC #1, CMDBLK, 20\$	

00000000G	EF	00000000G	00	FB	00288	CALLS	#0, SPLIT	1117
			EF	9F	0028F	PUSHAB	STR\$FAILURE	
			7E	D4	00295	CLRL	-(SP)	
		03E8	58	DD	00297	PUSHL	R8	
			C7	9F	00299	PUSHAB	\$STR\$STRING	
			7E	D4	0029D	CLRL	-(SP)	
00000000G	EF		05	FB	0029F	CALLS	#5, XST\$COPY	
	50		68	3C	002A6	MOVZWL	TMPSTR, R0	
00000G00G	EF		50	C0	002A9	ADDL2	R0, CHROUT	
	69		68	9E	002B0	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
E8	A9		07	90	002B3	MOVB	#7, IOB\$+44	
			5B	DD	002B7	PUSHL	R11	
			7E	D4	002B9	CLRL	-(SP)	
		BC	A9	9F	002BB	PUSHAB	IOB\$	
00000000G	EF		03	FB	002BE	CALLS	#3, XPOS\$PUT	
	6A		01	E1	002C5	BBC	#1, CMDBLK, 21\$	
00000000G	EF	00000000G	00	FB	002C9	CALLS	#0, SPLIT	1118
			EF	9F	002D0	PUSHAB	STR\$FAILURE	
			7E	D4	002D6	CLRL	-(SP)	
		03F0	58	DD	002D8	PUSHL	R8	
			C7	9F	002DA	PUSHAB	\$STR\$STRING	
			7E	D4	002DE	CLRL	-(SP)	
00000000G	EF		05	FB	002E0	CALLS	#5, XST\$COPY	
	50		68	3C	002E7	MOVZWL	TMPSTR, R0	
00000000G	EF		50	C0	002EA	ADDL2	R0, CHROUT	
	69		68	9E	002F1	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
E8	A9		07	90	002F4	MOVB	#7, IOB\$+44	
			5B	DD	002F8	PUSHL	R11	
			7E	D4	002FA	CLRL	-(SP)	
		BC	A9	9F	002FC	PUSHAB	IOB\$	
00000000G	EF		03	FB	002FF	CALLS	#3, XPOS\$PUT	
	6A		01	E1	00306	BBC	#1, CMDBLK, 23\$	
00000000G	EF	00000000G	00	FB	0030A	CALLS	#0, SPLIT	1121
		03A4	EF	D4	00311	CLRL	HL N	1122
			C7	D4	00317	CLRL	LASTHL	1059
			3A	11	0031B	BRB	28\$	1127
53	00000000G		EF	D0	0031D	MOVL	MINOR, R3	
			50	D4	00324	CLRL	R0	
07			53	D1	00326	CMPL	R3, #7	
			02	12	00329	BNEQ	25\$	
			50	D6	0032B	INCL	R0	
			51	D4	0032D	CLRL	R1	1128
0D			53	D1	0032F	CMPL	R3, #13	
			02	12	00332	BNEQ	26\$	
			51	D6	00334	INCL	R1	
51			50	C8	00336	BISL2	R0, R1	1129
			50	D4	00339	CLRL	R0	
0A			53	D1	0033B	CMPL	R3, #10	
			02	12	0033E	BNEQ	27\$	
			50	D6	00340	INCL	R0	
50			51	C8	00342	BISL2	R1, R0	1126
55			52	D2	00345	MCOML	R2, R5	
50			55	CA	00348	BICL2	R5, R0	
01			50	D1	0034B	CMPL	R0, #1	
			09	12	0034E	BNEQ	29\$	
00000000V	EF		00	FB	00350	CALLS	#0, PROCESS_ENTITY_INFO	1135
			5E	11	00357	BRB	36\$	

51	00000000G	EF	D0	00359	29\$:	MOVL	MINOR, R1	1139
08		53	D4	00360		CLRL	R3	
		51	D1	00362		CMPL	R1, #8	
		02	12	00365		BNEQ	30\$	
		53	D6	00367		INCL	R3	
09		50	D4	00369	30\$:	CLRL	R0	1140
		51	D1	00368		CMPL	R1, #9	
		02	12	0036E		BNEQ	31\$	
		50	D6	00370		INCL	R0	
50		53	C8	00372	31\$:	BISL2	R3, R0	
08		53	D4	00375		CLRL	R3	1141
		51	D1	00377		CMPL	R1, #11	
		02	12	0037A		BNEQ	32\$	
		53	D6	0037C		INCL	R3	
53		50	C8	0037E	32\$:	BISL2	R0, R3	
0C		50	D4	00381		CLRL	R0	1142
		51	D1	00383		CMPL	R1, #12	
		02	12	00386		BNEQ	33\$	
		50	D6	00388		INCL	R0	
50		53	C8	0038A	33\$:	BISL2	R3, R0	
0E		53	D4	0038D		CLRL	R3	1143
		51	D1	0038F		CMPL	R1, #14	
		02	12	00392		BNEQ	34\$	
		53	D6	00394		INCL	R3	
53		50	C8	00396	34\$:	BISL2	R0, R3	
0F		50	D4	00399		CLRL	R0	1144
		51	D1	0039B		CMPL	R1, #15	
		02	12	0039E		BNEQ	35\$	
		50	D6	003A0		INCL	R0	
50		53	C8	003A2	35\$:	BISL2	R3, R0	
55		52	D2	003A5		MCOML	R2, R5	1138
50		55	CA	003A8		BICL2	R5, R0	
01		50	D1	003AB		CMPL	R0, #1	
		0A	12	003AE		BNEQ	37\$	
00000000V	EF	00	FB	003B0		CALLS	#0, PROCESS_ENTITY_TXT	1150
		FC7B	31	003B7	36\$:	BRW	1\$	
		50	D4	003BA	37\$:	CLRL	R0	1152
	0CJ00000G	EF	D5	003BC		TSTL	MINOR	
		02	12	003C2		BNEQ	38\$	
		50	D6	003C4		INCL	R0	
51		52	D2	003C6	38\$:	MCOML	R2, R1	
50		51	CA	003C9		BICL2	R1, R0	
01		50	D1	003CC		CMPL	R0, #1	
		03	13	003CF		BEQL	39\$	
		00AF	31	003D1		BRW	44\$	
	03B4	C7	D4	003D4	39\$:	CLRL	ERRCNT	1157
		04	B0	003D8		MOVW	#4, IOB\$+52	1161
02	A7	02	90	003DB		MOVB	#2, IOB\$+54	
F8	A7	06	90	003DF		MOVB	#6, IOB\$+44	
		5B	DD	003E3		PUSHL	R11	
		7E	D4	003E5		CLRL	-(SP)	
		CC	A7	9F	003E7	PUSHAB	IOB\$	
00000000G	EF	03	FB	003EA		CALLS	#3, XPOSGET	
03A0	C7	04	B7	003F1		MOVL	@TOC IOB\$+56, HLCNT	1162
	67	04	B0	003F7		MOVW	#4, IOB\$+52	1166
02	A7	02	90	003FA		MOVB	#2, IOB\$+54	
F8	A7	06	90	003FE		MOVB	#6, IOB\$+44	

				5B	DD	00402	PUSHL	R11		
				7E	D4	00404	CLRL	-(SP)		
			CC	A7	9F	00406	PUSHAB	IOB\$		
				03	FB	00409	CALLS	#3, XPOS\$GET		
	00000000G	EF		B7	D0	00410	MOVL	@TOCIOB+56, HL_N		1167
67	00000000G	EF	04	04	A5	00418	MULW3	#4, HLCNT, IOB\$+52		1172
	03A0	C7		02	90	0041E	MOVB	#2, IOB\$+54		
	02	A7		06	90	00422	MOVB	#6, IOB\$+44		
	F8	A7		5B	DD	00426	PUSHL	R11		
				7E	D4	00428	CLRL	-(SP)		
			CC	A7	9F	0042A	PUSHAB	IOB\$		
	00000000G	EF		03	FB	0042D	CALLS	#3, XPOS\$GET		
				50	D4	00434	CLRL	I		1178
				0E	11	00436	BRB	41\$		
			04	B740	DE	00438	MOVAL	@TOCIOB+56[I], R1		1183
	00000000G	EF	FC	A1	D0	0043D	MOVL	-4(R1), HLLIST-4[I]		
EC			03A0	C7	F3	00446	AOBLEQ	HLCNT, I, 40\$		1178
67	03A0	C7		04	A5	0044C	MULW3	#4, HLCNT, IOB\$+52		1190
	02	A7		02	90	00452	MOVB	#2, IOB\$+54		
	F8	A7		06	90	00456	MOVB	#6, IOB\$+44		
				5B	DD	0045A	PUSHL	R11		
				7E	D4	0045C	CLRL	-(SP)		
			CC	A7	9F	0045E	PUSHAB	IOB\$		
	00000000G	EF		03	FB	00461	CALLS	#3, XPOS\$GET		
				50	D4	00468	CLRL	I		1196
				0E	11	0046A	BRB	43\$		
			04	B740	DE	0046C	MOVAL	@TOCIOB+56[I], R1		1201
	00000000G	EF	FC	A1	D0	00471	MOVL	-4(R1), HLDSP-4[I]		
EC			03A0	C7	F3	0047A	AOBLEQ	HLCNT, I, 42\$		1196
				FBB2	31	00480	BRW	1\$		1059
				51	D4	00483	CLRL	R1		1209
				EF	D1	00485	CMPL	MINOR, #2		
	02	00000000G		02	12	0048C	BNEQ	45\$		
				51	D6	0048E	INCL	R1		
				50	D4	00490	CLRL	R0		1210
	06	00000000G		EF	D1	00492	CMPL	MINOR, #6		
				02	12	00499	BNEQ	46\$		
				50	D6	0049B	INCL	R0		
	50			51	C8	0049D	BISL2	R1, R0		
	53			52	D2	004A0	MCOML	R2, R3		1208
	50			53	CA	004A3	BICL2	R3, R0		
	01			50	D1	004A6	CMPL	R0, #1		
				03	13	004A9	BEQL	47\$		
			01	C7	31	004AB	BRW	55\$		
			03B4	C7	D4	004AE	CLRL	ERRCNT		1220
				04	B0	004B2	MOVW	#4, IOB\$+52		1225
				02	90	004B5	MOVB	#2, IOB\$+54		
	02	A7		06	90	004B9	MOVB	#6, IOB\$+44		
	F8	A7		5B	DD	004BD	PUSHL	R11		
				7E	D4	004BF	CLRL	-(SP)		
			CC	A7	9F	004C1	PUSHAB	IOB\$		
	00000000G	EF		03	FB	004C4	CALLS	#3, XPOS\$GET		
	00000000G	EF	04	B7	D0	004CB	MOVL	@TOCIOB+56, TXTLEN		1226
				EF	D0	004D3	MOVL	TXTLEN, R0		1232
	50	00000000G		20	13	004DA	BEQL	48\$		
				03	C0	004DC	ADDL2	#3, R0		1234
	50			04	C6	004DF	DIVL2	#4, R0		

67	02	50	04	A5	004E2	MULW3	#4, R0, IOB\$+52	
	F8	A7	02	90	004E6	MOVB	#2, IOB\$+54	
			06	90	004EA	MOVB	#6, IOB\$+44	
			5B	DD	004EE	PUSHL	R11	
			7E	D4	004F0	CLRL	-(SP)	
		CC	A7	9F	004F2	PUSHAB	IOB\$	
00000000G	EF		03	FB	004F5	CALLS	#3, XPOSGET	
04	AA	00000000G	EF	D1	004FC	CMPL	HL N, CMDBLK+4	1240
			03	15	00504	BLEQ	49\$	
			0160	31	00506	BRW	54\$	
	1C	AA	EF	D1	00509	CMPL	HL N, CMDBLK+28	1244
41			45	14	00511	BGTR	50\$	
	6A	00000000G	01	E0	00513	BBS	#1, CMDBLK, 50\$	1245
			EF	9F	00517	PUSHAB	STR\$FAILURE	1247
			7E	D4	0051D	CLRL	-(SP)	
			5B	DD	0051F	PUSHL	R8	
		03F8	C7	9F	00521	PUSHAB	\$STR\$STRING	
			7E	D4	00525	CLRL	-(SP)	
00000000G	EF		05	FB	00527	CALLS	#5, XST\$COPY	
	50		68	3C	0052E	MOVZWL	TMPSTR, R0	
00000000G	EF		50	C0	00531	ADDL2	R0, CHROUT	
	69		68	9E	00538	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
E8	A9		07	90	0053B	MOVB	#7, IOB\$+44	
			5B	DD	0053F	PUSHL	R11	
			7E	D4	00541	CLRL	-(SP)	
		BC	A9	9F	00543	PUSHAB	IOB\$	
00000000G	EF		03	FB	00546	CALLS	#3, XPOSPUT	
07	6A		01	E1	0054D	BBC	#1, CMDBLK, 50\$	
00000000G	EF		00	FB	00551	CALLS	#0, SPLIT	
00000000G	EF	00000000G	EF	9E	00558	MOVAB	LINE, LP	1252
		00000000G	EF	D4	00563	CLRL	INTLIN	
		00000000G	EF	D4	00569	CLRL	EXTLIN	
00000000G	EF	20	AA	D0	0056F	MOVL	INDENTS, EXTLIN	1260
00000000G	EF	20	AA	D0	00577	MOVL	INDENTS, LINE_INDENT	1264
00000000V	EF		00	FB	0057F	CALLS	#0, OUTSEC	1269
00000000G	EF	00000000G	EF	D0	00586	MOVL	EXTLIN, WRAP	1273
00000000G	EF	00000000G	EF	9E	00591	MOVAB	TXTPAG, PTRPAG	1279
18	6A		03	E0	0059C	BBS	#3, CMDBLK, 51\$	1281
			0A	DD	005A0	PUSHL	#10	1288
		00000000G	EF	9F	005A2	PUSHAB	PTRPAG	
	7E	00000000G	EF	3C	005A8	MOVZWL	PAGEN+14, -(SP)	
00000000G	EF		03	FB	005AF	CALLS	#3, PACBAS	
			13	11	005B6	BRB	52\$	
		00000000G	EF	9F	005B8	PUSHAB	PTRPAG	1295
		00000000G	EF	9F	005BE	PUSHAB	PAGEN	
00000000G	EF		02	FB	005C4	CALLS	#2, PACPAG	
00000000G	EF		50	D0	005CB	MOVL	R0, LENPAG	
50	18	AA	EF	C3	005D2	SUBI3	LENPAG, CMDBLK+24, R0	1301
00000000G	EF	FF	A0	9E	005DB	MOVAB	-1(R0), RMARGIN	
		04	A7	DD	005E3	PUSHL	TOCIOB+56	1305
		00000000G	EF	DD	005E6	PUSHL	TXLEN	
00000000G	EF		02	FB	005EC	CALLS	#2, FMTTXT	
00000000G	EF		00	FB	005F3	CALLS	#0, INSREF	1309
	OC	AE	EF	B0	005FA	MOVW	INTLIN, \$STR\$STRING	1313
	OE	AE	OE	90	00602	MOVB	#14, \$STR\$STRING+2	
	OF	AE	01	90	00606	MOVB	#1, \$STR\$STRING+3	
	10	AE	EF	9E	0060A	MOVAB	LINE, \$STR\$STRING+4	

		00000000G	EF	9F	00612	PUSHAB	STR\$FAILURE		
			7E	D4	00618	CLRL	-(SP)		
			58	DD	0061A	PUSHL	R8		
		18	AE	9F	0061C	PUSHAB	\$STR\$STRING		
			7E	D4	0061F	CLRL	-(SP)		
		00000000G	EF	05	FB	00621	CALLS	#5, XST\$COPY	
			50	68	3C	00628	MOVZWL	TMPSTR, R0	
		00000000G	EF	50	C0	0062B	ADDL2	R0, CHROUT	
			69	68	9E	00632	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		E8	A9	07	90	00635	MOVB	#7, IOB\$+44	
				58	DD	00639	PUSHL	R11	
			7E	D4	0063B	CLRL	-(SP)		
			BC	A9	9F	0063D	PUSHAB	IOB\$	
		00000000G	EF	03	FB	00640	CALLS	#3, XPOS\$PUT	
07			6A	01	E1	00647	BBC	#1, CMDBLK, 53\$	
		00000000G	EF	00	FB	0064B	CALLS	#0, SPLIT	
		00000000G	EF	00000000G	EF	9E	00652	MOVAB	LINE, LP
				00000000G	EF	D4	0065D	CLRL	INTLIN
				00000000G	EF	D4	00663	CLRL	EXTLIN
		03A4	C7	00000000G	EF	D0	00669	MOVL	HL_N, LASTHL
				F9C0	31	00672	BRW	1\$	
				51	D4	00675	CLRL	R1	
		10	00000000G	EF	D1	00677	CMPL	MINOR, #16	
				02	12	0067E	BNEQ	56\$	
				51	D6	00680	INCL	R1	
				50	D4	00682	CLRL	R0	
		13	00000000G	EF	D1	00684	CMPL	MINOR, #19	
				02	12	0068B	BNEQ	57\$	
				50	D6	0068D	INCL	R0	
		50		51	C8	0068F	BISL2	R1, R0	
		53		52	D2	00692	MCOML	R2, R3	
		50		53	CA	00695	BICL2	R3, R0	
		01		50	D1	00698	CMPL	R0, #1	
				03	13	0069B	BEQL	58\$	
				0111	31	0069D	BRW	61\$	
		02	67	04	B0	006A0	MOVW	#4, IOB\$+52	
		F8	A7	02	90	006A3	MOVB	#2, IOB\$+54	
				06	90	006A7	MOVB	#6, IOB\$+44	
				58	DD	006AB	PUSHL	R11	
				7E	D4	006AD	CLRL	-(SP)	
			CC	A7	9F	006AF	PUSHAB	IOB\$	
		00000000G	EF	03	FB	006B2	CALLS	#3, XPOS\$GET	
		03A8	C7	B7	D0	006B9	MOVL	@TOCIOB+56, TOC_PAGE	
			67	04	B0	006BF	MOVW	#4, IOB\$+52	
		02	A7	02	90	006C2	MOVB	#2, IOB\$+54	
		F8	A7	06	90	006C6	MOVB	#6, IOB\$+44	
				58	DD	006CA	PUSHL	R11	
				7E	D4	006CC	CLRL	-(SP)	
			CC	A7	9F	006CE	PUSHAB	IOB\$	
		00000000G	EF	03	FB	006D1	CALLS	#3, XPOS\$GET	
			67	04	B0	006D8	MOVW	#4, IOB\$+52	
		02	A7	02	90	006DB	MOVB	#2, IOB\$+54	
		F8	A7	06	90	006DF	MOVB	#6, IOB\$+44	
				58	DD	006E3	PUSHL	R11	
				7E	D4	006E5	CLRL	-(SP)	
			CC	A7	9F	006E7	PUSHAB	IOB\$	
		00000000G	EF	03	FB	006EA	CALLS	#3, XPOS\$GET	

1314

1317

1059

1322

1323

1321

1333

1334

1335

1336

67	02	52	04	B7	D0	006F1	MOVL	@TOC10B+56, LEN	1337
		50	03	21	15	006F5	BLEQ	59\$	1339
		50		A2	9E	006F7	MOVAB	3(R2), R0	1341
		50		04	C6	006FB	DIVL2	#4, R0	
		50		04	A5	006FE	MULW3	#4, R0, 10B\$+52	
	02	A7		02	90	00702	MOVB	#2, 10B\$+54	
	F8	A7		06	90	00706	MOVB	#6, 10B\$+44	
				5B	DD	0070A	PUSHL	R11	
				7E	D4	0070C	CLRL	-(SP)	
			CC	A7	9F	0070E	PUSHAB	10B\$	
	00000000G	EF		03	FB	00711	CALLS	#3, XPOSGET	
	OC	AE		52	B0	00718	MOVW	LEN, \$STR\$STRING	1344
	OE	AE		0E	90	0071C	MOVB	#14, \$STR\$STRING+2	
	OF	AE		01	90	00720	MOVB	#1, \$STR\$STRING+3	
	10	AE	04	A7	D0	00724	MOVL	TOC10B+56, \$STR\$STRING+4	
			00000000G	EF	9F	00729	PUSHAB	STR\$FAILURE	
				7E	D4	0072F	CLRL	-(SP)	
			00000000G	EF	9F	00731	PUSHAB	\$STR\$TARGET	
			18	AE	9F	00737	PUSHAB	\$STR\$STRING	
				7E	D4	0073A	CLRL	-(SP)	
	00000000G	EF		05	FB	0073C	CALLS	#5, XST\$COPY	
		67		04	B0	00743	MOVW	#4, 10B\$+52	1346
	02	A7		02	90	00746	MOVB	#2, 10B\$+54	
	F8	A7		06	90	0074A	MOVB	#6, 10B\$+44	
				5B	DD	0074E	PUSHL	R11	
				7E	D4	00750	CLRL	-(SP)	
			CC	A7	9F	00752	PUSHAB	10B\$	
	00000000G	EF		03	FB	00755	CALLS	#3, XPOSGET	
		52	04	B7	D0	0075C	MOVL	@TOC10B+56, LEN	1347
				21	15	00760	BLEQ	60\$	1349
		50	03	A2	9E	00762	MOVAB	3(R2), R0	1351
		50		04	C6	00766	DIVL2	#4, R0	
		50		04	A5	00769	MULW3	#4, R0, 10B\$+52	
	02	A7		02	90	0076D	MOVB	#2, 10B\$+54	
	F8	A7		06	90	00771	MOVB	#6, 10B\$+44	
				5B	DD	00775	PUSHL	R11	
				7E	D4	00777	CLRL	-(SP)	
			CC	A7	9F	00779	PUSHAB	10B\$	
	00000000G	EF		03	FB	0077C	CALLS	#3, XPOSGET	
	OC	AE		52	B0	00783	MOVW	LEN, \$STR\$STRING	1354
	OE	AE		0E	90	00787	MOVB	#14, \$STR\$STRING+2	
	OF	AE		01	90	0078B	MOVB	#1, \$STR\$STRING+3	
	10	AE	04	A7	D0	0078F	MOVL	TOC10B+56, \$STR\$STRING+4	
			00000000G	EF	9F	00794	PUSHAB	STR\$FAILURE	
				7E	D4	0079A	CLRL	-(SP)	
			00000000G	EF	9F	0079C	PUSHAB	\$STR\$TARGET	
			18	AE	9F	007A2	PUSHAB	\$STR\$STRING	
				7E	D4	007A5	CLRL	-(SP)	
	00000000G	EF		05	FB	007A7	CALLS	#5, XST\$COPY	
			F884	31	007AE	BRW	1\$		1059
		51	00000000G	EF	D0	007B1	MOVL	MINOK ?1	1360
				53	D4	007B8	CLRL	R3	
		03		51	D1	007BA	CMPL	R1, #3	
				02	12	007BD	BNEQ	62\$	
				53	D6	007BF	INCL	R3	
				50	D4	007C1	CLRL	R0	1361
		11		51	D1	007C3	CMPL	R1, #17	

			02	12	007C6	BNEQ	63\$		
			50	D6	007C8	INCL	R0		
			53	C8	007CA	BISL2	R3, R0		
			53	D4	007CD	CLRL	R3		1362
			12	D1	007CF	CMPL	R1, #18		
			02	12	007D2	BNEQ	64\$		
			53	D6	007D4	INCL	R3		
			53	C8	007D6	BISL2	R0, R3		
			50	D4	007D9	CLRL	R0		1363
			04	D1	007DB	CMPL	R1, #4		
			02	12	007DE	BNEQ	65\$		
			50	D6	007E0	INCL	R0		
			50	C8	007E2	BISL2	R3, R0		
			53	D4	007E5	CLRL	R3		1364
			14	D1	007E7	CMPL	R1, #20		
			02	12	007EA	BNEQ	66\$		
			53	D6	007EC	INCL	R3		
			53	C8	007EE	BISL2	R0, R3		
			50	D4	007F1	CLRL	R0		1365
			15	D1	007F3	CMPL	R1, #21		
			02	12	007F6	BNEQ	67\$		
			50	D6	007F8	INCL	R0		
			50	C8	007FA	BISL2	R3, R0		
			55	D2	007FD	MCOML	R2, R5		1359
			50	CA	00800	BICL2	R5, R0		
			01	D1	00803	CMPL	R0, #1		
			03	13	00806	BEQL	68\$		
			04C7	31	00808	BRW	104\$		
			03B4	C7	D4	0080B	CLRL	ERRCNT	1375
			00000000G	EF	9E	0080F	MOVAB	LINE, LP	1376
			00000000G	EF	D4	0081A	CLRL	INTLIN	
			00000000G	EF	D4	00820	CLRL	EXTLIN	
03			6A	01	E0	00826	BBS	#1, CMDBLK, 69\$	1378
			00C8	31	0082A	BRW	74\$		
			00000000G	EF	9F	0082D	PUSHAB	STR\$FAILURE	1384
			7E	D4	00833	CLRL	-(SP)		
			58	DD	00835	PUSHL	R8		
			0400	C7	9F	00837	PUSHAB	\$STR\$STRING	
			7E	D4	0083B	CLRL	-(SP)		
			00000000G	EF	05	FB	0083D	CALLS	#5, XST\$COPY
			50	3C	00844	MOVZWL	TMP\$R, R0		
			00000000G	EF	50	C0	00847	ADDL2	R0, CHROUT
			69	9E	0084E	MOVAB	\$JOB\$OUTPUT, 10B\$+68		
			E8	A9	07	90	00851	MOVAB	#7, 10B\$+44
				5B	DD	00855	PUSHL	R11	
				7E	D4	00857	CLRL	-(SP)	
			BC	A9	9F	00859	PUSHAB	10B\$	
			00000000G	EF	03	FB	0085C	CALLS	#3, XPOS\$PUT
07			6A	01	E1	00863	BBC	#1, CMDBLK, 70\$	
			00000000G	EF	00	FB	00867	CALLS	#0, SPLIT
			00000000G	EF	D5	0086E	TSTL	HL N	1386
			41	13	00874	BEQL	71\$		
			00000000G	EF	9F	00876	PUSHAB	STR\$FAILURE	
			7E	D4	0087C	CLRL	-(SP)		
			58	DD	0087E	PUSHL	R8		
			0408	C7	9F	00880	PUSHAB	\$STR\$STRING	
				7E	D4	00884	CLRL	-(SP)	

	00000000G	EF	05	FB	00886	CALLS	#5, XST\$COPY		
		50	68	3C	0088D	MOVZWL	TMPSTR, R0		
	00000000G	EF	50	C0	00890	ADDL2	R0, CHROUT		
		69	68	9E	00897	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
	E8	A9	07	90	0089A	MOVB	#7, IOB\$+44		
			5B	DD	0089E	PUSHL	R11		
			7E	D4	008A0	CLRL	-(SP)		
			A9	9F	008A2	PUSHAB	IOB\$		
	00000000G	EF	03	FB	008A5	CALLS	#3, XPOS\$PUT		
07		6A	01	E1	008AC	BBC	#1, CMDBLK, 71\$		
	00000000G	EF	00	FB	008B0	CALLS	#0, SPLIT		
		1C	C7	E8	008B7	BLBS	TOC PAGE, 72\$		1388
00000000G	FF	00000000'	05	28	008BC	MOV3	#5, P.AAN, @LP		1390
		00000000G	05	C0	008C8	ADDL2	#5, LP		
		00000000G	05	C0	008CF	ADDL2	#5, INTLIN		
			1A	11	008D6	BRB	73\$		1388
00000000G	FF	00000000'	06	28	008D8	MOV3	#6, P.AAO, @LP		1392
		00000000G	06	C0	008E4	ADDL2	#6, LP		
		00000000G	06	C0	008EB	ADDL2	#6, INTLIN		
			0111	31	008F2	BRW	80\$		1378
			EF	9F	008F5	PUSHAB	STR\$FAILURE		1400
			7E	D4	008FB	CLRL	-(SP)		
			5B	DD	008FD	PUSHL	R8		
			C7	9F	008FF	PUSHAB	\$STR\$STRING		
			7E	D4	00903	CLRL	-(SP)		
	00000000G	EF	05	FB	00905	CALLS	#5, XST\$COPY		
		50	68	3C	0090C	MOVZWL	TMPSTR, R0		
	00000000G	EF	50	C0	0090F	ADDL2	R0, CHROUT		
		69	68	9E	00916	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
	E8	A9	07	90	00919	MOVB	#7, IOB\$+44		
			5B	DD	0091D	PUSHL	R11		
			7E	D4	0091F	CLRL	-(SP)		
			A9	9F	00921	PUSHAB	IOB\$		
	00000000G	EF	03	FB	00924	CALLS	#3, XPOS\$PUT		
07		6A	01	E1	0092B	BBC	#1, CMDBLK, 75\$		
	00000000G	EF	00	FB	0092F	CALLS	#0, SPLIT		
			AA	D5	00936	TSTL	CMDBLK+4		1402
			59	13	00939	BEQL	78\$		
			EF	D5	0093B	TSTL	HL N		1406
			10	13	00941	BEQL	76\$		
			EF	9F	00943	PUSHAB	STR\$FAILURE		
			7E	D4	00949	CLRL	-(SP)		
			5B	DD	0094B	PUSHL	R8		
			C7	9F	0094D	PUSHAB	\$STR\$STRING		
			0E	11	00951	BRB	77\$		
			EF	9F	00953	PUSHAB	STR\$FAILURE		
			7E	D4	00959	CLRL	-(SP)		
			5B	DD	0095B	PUSHL	R8		
			C7	9F	0095D	PUSHAB	\$STR\$STRING		
			7E	D4	00961	CLRL	-(SP)		
	00000000G	EF	05	FB	00963	CALLS	#5, XST\$COPY		
		50	68	3C	0096A	MOVZWL	TMPSTR, R0		
	00000000G	EF	50	C0	0096D	ADDL2	R0, CHROUT		
		69	68	9E	00974	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
	E8	A9	07	90	00977	MOVB	#7, IOB\$+44		
			5B	DD	0097B	PUSHL	R11		
			7E	D4	0097D	CLRL	-(SP)		

07	00000000G	EF	BC	A9	9F	0097F	PUSHAB	IOB\$	
		6A		03	FB	00982	CALLS	#3, XPO\$PUT	
	00000000G	EF		01	E1	00989	BBC	#1, CMDBLK, 78\$	
		50	20	00	FB	0098D	CALLS	#0, SPLIT	
				AA	D0	00994	MOVL	INDENTS, R0	1410
				5C	13	00998	BEQL	79\$	
				7E	D4	0099A	CLRL	-(SP)	1412
				50	DD	0099C	PUSHL	R0	
	00000000G	7E	0903	8F	3C	0099E	MOVZWL	#2307, -(SP)	
		EF		03	FB	009A3	CALLS	#3, XST\$ASCII	
				50	DD	009AA	PUSHL	R0	
	00000000G	EF	0428	C7	9F	009AC	PUSHAB	\$STR\$STRINGO	
				02	FB	009B0	CALLS	#2, XST\$JOIN	
			00000000G	EF	9F	009B7	PUSHAB	STR\$FAILURE	
				7E	D4	009BD	CLRL	-(SP)	
			0101	8F	BB	009BF	PUSHR	#M<R0,R8>	
				7E	D4	009C3	CLRL	-(SP)	
	00000000G	EF		05	FB	009C5	CALLS	#5, XST\$COPY	
		50		68	3C	009CC	MOVZWL	TMPSTR, R0	
	00000000G	EF		50	C0	009CF	ADDL2	R0, CHROUT	
		69		68	9E	009D6	MOVAB	\$IOB\$OUTPUT, IOB:+68	
	E8	A9		07	90	009D9	MOVB	#7, IOB\$+44	
				5B	DD	009DD	PUSHL	R11	
				7E	D4	009DF	CLRL	-(SP)	
			EC	A9	9F	009E1	PUSHAB	IOB\$	
07	00000000G	EF		03	FB	009E4	CALLS	#3, XPO\$PUT	
		6A		01	E1	009EB	BBC	#1, CMDBLK, 79\$	
	00000000G	EF		00	FB	009EF	CALLS	#0, SPLIT	
	00000000G	EF	20	AA	D0	009F6	MOVL	INDENTS, LINE INDENT	1414
	00000000G	EF	18	AA	D0	009FE	MOVL	CMDBLK+24, RMARGIN	1415
			03A4	C7	D4	00A06	CLRL	LASTHL	1421
			00000000G	EF	D4	00A0A	CLRL	HL N	1422
		50	00000000G	EF	D0	00A10	MOVL	MINOR, R0	1424
		12		50	D1	00A17	CMPL	R0, #18	
				03	13	00A1A	BEQL	81\$	
		15		50	D1	00A1C	CMPL	R0, #21	1425
				03	12	00A1F	BNEQ	82\$	
			00C0	31	00A21	BRW	92\$		
		03		50	D1	00A24	CMPL	R0, #3	1431
				05	13	00A27	BEQL	83\$	
		11		50	D1	00A29	CMPL	R0, #17	1432
				16	12	00A2C	BNEQ	86\$	
08		6A		01	E1	00A2E	BBC	#1, CMDBLK, 84\$	1438
			00000000'	EF	9F	00A32	PUSHAB	P.AAV	1440
				06	11	00A38	BRB	85\$	
			00000000'	EF	9F	00A3A	PUSHAB	P.AAW	1442
				08	DD	00A40	PUSHL	#8	
				14	11	00A42	BRB	89\$	
08		6A		01	E1	00A44	BBC	#1, CMDBLK, 87\$	1450
			00000000'	EF	9F	00A48	PUSHAB	P.AAX	1452
				06	11	00A4E	BRB	88\$	
			00000000'	EF	9F	00A50	PUSHAB	P.AAY	1454
				09	DD	00A56	PUSHL	#9	
	00000000G	EF		02	FB	00A58	CALLS	#2, FMTTXT	
		50	00000000G	EF	3C	00A5F	MOVZWL	PRESTR, R0	1461
				0F	13	00A66	BEQL	90\$	
			00000000G	EF	DD	00A68	PUSHL	PRESTR+4	1463

				50 DD 00A6E	PUSHL R0		
				02 FB 00A70	CALLS #2, FMTTXT		
				EF 9E 00A77 90\$:	MOVAB TXISCT, PTRSCT	1468	
				EF 9F 00A82	PUSHAB PTRSCT	1469	
				EF 9F 00A88	PUSHAB PAGEN		
				02 FB 00A8E	CALLS #2, PACSEC		
				50 D0 00A95	MOVL R0, LENSCT		
				EF 9F 00A9C	PUSHAB TXISCT	1470	
				EF DD 00AA2	PUSHL LENSCT		
				02 FB 00AA8	CALLS #2, FMTTXT		
				50 00000000G EF 3C 00AAF	MOVZWL SFXSTR, R0	1475	
				0F 13 00AB6	BEQL 91\$		
				EF DD 00AB8	PUSHL SFXSTR+4	1477	
				50 DD 00ABE	PUSHL R0		
				02 FB 00AC0	CALLS #2, FMTTXT		
10				01 E0 00AC7 91\$:	BBS #1, CMDBLK, 93\$	1479	
				20 90 00ACB	MOVB #32, @LP		
				EF D6 00AD2	INCL LP		
				EF D6 00AD8	INCL INTLIN		
				EF D6 00ADE	INCL EXTLIN		
14				01 E1 00AE4 92\$:	BBC #1, CMDBLK, 94\$	1482	
				8F 90 00AE8 93\$:	MOVB #64, @LP		
				EF D6 00AF0	INCL LP		
				EF D6 00AF6	INCL INTLIN		
				04 B0 00AFC 94\$:	MOVW #4, IOB\$+52	1488	
				02 90 00AFF	MOVB #2, IOB\$+54		
				06 90 00B03	MOVB #6, IOB\$+44		
				5B DD 00B07	PUSHL R11		
				7E D4 00B09	CLRL -(SP)		
				A7 9F 00B0B	PUSHAB IOB\$		
				03 FB 00B0E	CALLS #3, XPOSGET		
				B7 D0 00B15	MOVL @TOCIOB+56, TXTLEN	1489	
				EF D0 00B1D	MOVL TXTLEN, R0	1494	
				20 15 00B24	BLEQ 95\$		
				03 C0 00B26	ADDL2 #3, R0	1496	
				04 C6 00B29	DIVL2 #4, R0		
67				04 A5 00B2C	MULW3 #4, R0, IOB\$+52		
				02 90 00B30	MOVB #2, IOB\$+54		
				06 90 00B34	MOVB #6, IOB\$+44		
				5B DD 00B38	PUSHL R11		
				7E D4 00B3A	CLRL -(SP)		
				A7 9F 00B3C	PUSHAB IOB\$		
				03 FB 00B3F	CALLS #3, XPOSGET		
35				01 E0 00B46 95\$:	BBS #1, CMDBLK, 96\$	1498	
56				AA C1 00B4A	ADDL3 INDENTS+4, INDENTS, R6	1500	
				56 D1 00B50	CMPL R6, EXTLIN		
				26 15 00B57	BLEQ 96\$		
				EF C2 00B59	SUBL2 EXTLIN, R6		
56				00 2C 00B60	MOVCS #0, (SP), #32, R6, @LP		
				FF 00B65			
				56 C0 00B6A	ADDL2 R6, LP		
				56 C0 00B71	ADDL2 R6, INTLIN		
				56 C0 00B78	ADDL2 R6, EXTLIN		
				EF D0 00B7F 96\$:	MOVL EXTLIN, WRAP	1505	
				C7 E9 00B8A	BLBC TOC PAGE, 99\$	1507	
				EF 9E 00B8F	MOVAB TXIPAG, PTRPAG	1513	
18				03 E0 00B9A	BBS #3, CMDBLK, 97\$	1515	

			0A DD 00B9E	PUSHL #10	1517
			EF 9F 00BA0	PUSHAB PTRPAG	
			EF 3C 00BA6	MOVZWL PAGEN+14, -(SP)	
			03 FB 00BAD	CALLS #3, PACBAS	
			13 11 00BB4	BRB 98\$	
			EF 9F 00BB6 97\$:	PUSHAB PTRPAG	1519
			EF 9F 00BBC	PUSHAB PAGEN	
			02 FB 00BC2	CALLS #2, PACPAG	
			50 D0 00BC9 98\$:	RO, LENPAG	
			EF C3 00BD0	SUBL3 LENPAG, CMDBLK+24, R0	1521
			A0 9E 00BD9	MOVAB -1(R0), RMARGIN	
			A7 DD 00BE1 99\$:	PUSHL TOC10B+56	1527
			EF DD 00BE4	PUSHL TXTLEN	
			02 FB 00BEA	CALLS #2, FMTTXT	
			C7 E9 00BF1	BLBC TOC_PAGE, 100\$	1529
			00 FB 00BF6	CALLS #0, INSREF	1535
			C7 D4 00BFD	CLRL TOC_PAGE	1536
			18 11 00C01	BRB 101\$	1529
			01 E1 00C03 100\$:	BBC #1, CMDBLK, 101\$	1541
			8F 90 00C07	MOVB #64, ALP	
			EF D6 00C0F	INLL LP	
			EF D6 00C15	INCL INTLIN	
			EF B0 00C1B 101\$:	MOVW INTLIN, \$STR\$STRING	1547
			0E 90 00C23	MOVB #14, \$STR\$STRING+2	
			01 90 00C27	MOVB #1, \$STR\$STRING+3	
			EF 9E 00C2B	MOVAB LINE, \$STR\$STRING+4	
			EF 9F 00C33	PUSHAB STR\$FAILURE	
			7E D4 00C39	CLRL -(SP)	
			58 DD 00C3B	PUSHL R8	
			AE 9F 00C3D	PUSHAB \$STR\$STRING	
			7E D4 00C40	CLRL -(SP)	
			05 FB 00C42	CALLS #5, XST\$COPY	
			68 3C 00C49	MOVZWL TMPSTR, R0	
			50 C0 00C4C	ADDL2 R0, CHROUT	
			68 9E 00C53	MOVAB \$10B\$OUTPUT, 10B\$+68	
			07 90 00C56	MOVB #7, 10B\$+44	
			58 DD 00C5A	PUSHL R11	
			7E D4 00C5C	CLRL -(SP)	
			A9 9F 00C5E	PUSHAB 10B\$	
			03 FB 00C61	CALLS #3, XPOS\$PUT	
			01 E1 00C68	BBC #1, CMDBLK, 102\$	
			00 FB 00C6C	CALLS #0, SPLIT	
			EF 9E 00C73 102\$:	MOVAB LINE, LP	1548
			EF D4 00C7E	CLRL INTLIN	
			EF D4 00C84	CLRL EXTIN	
			01 E0 00C8A	BBS #1, CMDBLK, 103\$	1552
			EF 9F 00C8E	PUSHAB STR\$FAILURE	
			7E D4 00C94	CLRL -(SP)	
			58 DD 00C96	PUSHL R8	
			C7 9F 00C98	PUSHAB \$STR\$STRING	
			7E D4 00C9C	CLRL -(SP)	
			05 FB 00C9E	CALLS #5, XST\$COPY	
			68 3C 00CA5	MOVZWL TMPSTR, R0	
			50 C0 00CA8	ADDL2 R0, CHROUT	
			68 9E 00CAF	MOVAB \$10B\$OUTPUT, 10B\$+68	
			07 90 00CB2	MOVB #7, 10B\$+44	
			58 DD 00CB6	PUSHL R11	

07	00000000G	EF	BC	7E	D4	00CB8	CLRL	-(SP)	
		6A		A9	9F	00CBA	PUSHAB	IOB\$	
	00000000G	EF		03	FB	00CBD	CALLS	#3, XPOSPUT	
				01	E1	00CC4	BBC	#1, CMDBLK, 103\$	
				00	FB	00CC8	CALLS	#0, SPLIT	
				363	31	00CCF	BRW	1\$	1059
				54	D1	00CD2	CMPL	R4, #4	1555
				03	13	00CD5	BEQL	105\$	
				02AB	31	00CD7	BRW	124\$	
				55	3C	00CDA	MOVZWL	MINOR, TYPE	1566
				54	3C	00CE1	MOVZWL	MINOR+2, POSITION	1567
				67	B0	00CE8	MOVW	#4, IOB\$+52	1568
	02	A7		02	90	00CEB	MOVB	#2, IOB\$+54	
	F8	A7		06	90	00CEF	MOVB	#6, IOB\$+44	
				5B	DD	00CF3	PUSHL	R11	
				7E	D4	00CF5	CLRL	-(SP)	
				A7	9F	00CF7	PUSHAB	IOB\$	
	00000000G	EF	CC	03	FB	00CFA	CALLS	#3, XPOSGET	
		52	04	B7	D0	00D01	MOVL	@TOCIOB+56, LEN	1569
				21	15	00D05	BLEQ	106\$	1571
				50	A2	9E 00D07	MOVAB	3(R2), R0	1573
				50	04	C6 00D0B	DIVL2	#4, R0	
67				50	A5	00D0E	MULW3	#4, R0, IOB\$+52	
	02	A7		02	90	00D12	MOVB	#2, IOB\$+54	
	F8	A7		06	90	00D16	MOVB	#6, IOB\$+44	
				5B	DD	00D1A	PUSHL	R11	
				7E	D4	00D1C	CLRL	-(SP)	
				A7	9F	00D1E	PUSHAB	IOB\$	
	00000000G	EF	CC	03	FB	00D21	CALLS	#3, XPOSGET	
		01		54	D1	00D28	CMPL	POSITION, #1	1578
				03	13	00D2B	BEQL	107\$	
				01BB	31	00D2D	BRW	118\$	
				53	D4	00D30	CLRL	IOB PTR	1583
				55	D1	00D32	CMPL	TYPE, #1	1585
				54	12	00D35	BNEQ	109\$	
				52	B0	00D37	MOVW	LEN, \$STR\$STRING	1591
	OC	AE		0E	90	00D3B	MOVB	#14, \$STR\$STRING+2	
	OE	AE		01	90	00D3F	MOVB	#1, \$STR\$STRING+3	
	OF	AE		A7	D0	00D43	MOVL	TOCIOB+56, \$STR\$STRING+4	
	10	AE	04	EF	9F	00D48	PUSHAB	STR\$FAILURE	
			00000000G	7E	D4	00D4E	CLRL	-(SP)	
				5B	DD	00D50	PUSHL	R8	
				18	AE	9F 00D52	PUSHAB	\$STR\$STRING	
				7E	D4	00D55	CLRL	-(SP)	
	00000000G	EF		05	FB	00D57	CALLS	#5, XST\$COPY	
		50		68	3C	00D5E	MOVZWL	TMPSTR, R0	
	00000000G	EF		50	C0	00D61	ADDL2	R0, CHROUT	
		69		68	9E	00D68	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	E8	A9		07	90	00D6B	MOVB	#7, IOB\$+44	
				5B	DD	00D6F	PUSHL	R11	
				7E	D4	00D71	CLRL	-(SP)	
				A9	9F	00D73	PUSHAB	IOB\$	
07	00000000G	EF	BC	03	FB	00D76	CALLS	#3, XPOSPUT	
		6A		01	E1	00D7D	BBC	#1, CMDBLK, 108\$	
	00000000G	EF		00	FB	00D81	CALLS	#0, SPLIT	
				0182	31	00D88	BRW	121\$	1585
	00000000G	EF		01	D0	00D8B	MOVL	#1, HL_N	1598

	02		55	D1	00D92	CMPL	TYPE, #2	1600
			05	12	00D95	BNEQ	110\$	
	03A8	C7	01	D0	00D97	MOVL	#1, TOC PAGE	
00000000G	EF		06	D0	00D9C	110\$:	MOVAB	1605
00000000G	EF	00000000G	EF	9E	00DA3		LINE, LP	1609
		00000000G	EF	D4	00DAE	CLRL	INTLIN	
		00000000G	EF	D4	00DB4	CLRL	EXTLIN	
00000000G	EF	20	AA	D0	00DBA	MOVL	INDENTS, EXTLIN	1610
00000000G	EF	20	AA	D0	00DC2	MOVL	INDENTS, LINE_INDENT	1611
00000000V	EF		00	FB	00DCA	CALLS	#0, OUTSEC	1615
00000000G	EF	00000000G	EF	D0	00DD1	MOVL	EXTLIN, WRAP	1616
	43	03A8	C7	E9	00DDC	BLBC	TOC PAGE, 113\$	1618
18	00000000G	EF	EF	9E	00DE1	MOVAB	TXTPAG, PTRPAG	1624
	6A	00000000G	03	E0	00DEC	BBS	#3, CMDBLK, 111\$	1626
			0A	DD	00DF0	PUSHL	#10	1628
		00000000G	EF	9F	00DF2	PUSHAB	PTRPAG	
00000000G	7E	00000000G	EF	3C	00DF8	MOVZWL	PAGEN+14, -(SP)	
	EF		03	FB	00DFF	CALLS	#3, PACBAS	
		00000000G	13	11	00E06	BRB	112\$	
		00000000G	EF	9F	00E08	111\$:	PUSHAB	1630
		00000000G	EF	9F	00E0E	PUSHAB	PAGEN	
00000000G	EF		02	FB	00E14	CALLS	#2, PACPAG	
00000000G	EF		50	D0	00E1B	112\$:	MOVL	1618
			06	11	00E22	BRB	114\$	1634
		00000000G	EF	D4	00E24	113\$:	CLRL	1636
50	18	AA	EF	C3	00E2A	114\$:	SUBL3	
00000000G	EF	FF	A0	9E	00E33	MOVAB	LENPAG, CMDBLK+24, R0	
		04	A7	DD	00E3B	PUSHL	-1(R0), RMARGIN	
			52	DD	00E3E	PUSHL	TOC10B+56	1637
00000000G	EF		02	FB	00E40	CALLS	LEN	
00000000G	OD	03A8	C7	E9	00E47	BLBC	#2, FMTTXT	1639
	EF		00	FB	00E4C	CALLS	TOC_PAGE, 115\$	1642
		03A8	C7	D4	00E53	CLRL	#0, -INSREF	1643
			18	11	00E57	BRB	TOC PAGE	1639
14	00000000G	6A	01	E1	00E59	115\$:	BBC	1648
		FF	8F	90	00E5D	MOVAB	#1, CMDBLK, 116\$	
			EF	D6	00E65	INCL	#64, @LP	
		00000000G	EF	D6	00E6B	INCL	LP	
	OC	AE	EF	B0	00E71	116\$:	INTLIN	1651
	OE	AE	0E	90	00E79	MOVW	INTLIN, \$STR\$STRING	
	OF	AE	01	90	00E7D	MOVAB	#14, \$STR\$STRING+2	
	10	AE	EF	9E	00E81	MOVAB	#1, \$STR\$STRING+3	
		00000000G	EF	9F	00E89	MOVAB	LINE, \$STR\$STRING+4	
		00000000G	7E	D4	00E8F	PUSHAB	STR\$FAILURE	
			58	DD	00E91	CLRL	-(SP)	
		18	AE	9F	00E93	PUSHL	R8	
			7E	D4	00E96	PUSHAB	\$STR\$STRING	
00000000G	EF		05	FB	00E98	CLRL	-(SP)	
00000000G	50		68	3C	00E9F	CALLS	#5, XST\$COPY	
	EF		50	C0	00EA2	MCVZWL	IMPSTR, R0	
	69		68	9E	00EA9	ADDL2	R0, CHROUT	
	E8	A9	07	90	00EAC	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
			5B	DD	00EB0	MOVAB	#7, IOB\$+44	
			7E	D4	00EB2	PUSHL	R11	
		BC	A9	9F	00EB4	CLRL	-(SP)	
			03	FB	00EB7	PUSHAB	IOB\$	
07	00000000G	EF	01	E1	00EBE	CALLS	#3, XPOSPUT	
	6A					BBC	#1, CMDBLK, 117\$	

00000000G	EF	00	FB	00EC2	CALLS	#0, SPLIT	1652
00000000G	EF	00000000G	EF	9E 00EC9	MOVAB	LINE, LP	1653
		00000000G	EF	D4 00ED4	CLRL	INTLIN	1575
		00000000G	EF	D4 00EDA	CLRL	EXTLIN	1658
03A4	C7	00000000G	EF	D0 00EE0	MOVL	HL N, LASTHL	1659
	02		22	11 00EE9	BRB	12T\$	1661
			54	D1 00EEB	CMPL	POSITION, #2	1662
	53	02AC	07	12 00EEE	BNEQ	119\$	1664
			C7	9E 00EFO	MOVAB	EXIOB, IOB_PTR	1665
	03		16	11 00EF5	BRB	121\$	1669
			54	D1 00EF7	CMPL	POSITION, #3	1670
	53	00C4	07	12 00EFA	BNEQ	120\$	1680
			C7	9E 00EFC	MOVAB	FGIOB, IOB_PTR	1681
	04		0A	11 00F01	BRB	121\$	1682
			54	D1 00F03	CMPL	POSITION, #4	1683
	53	01B8	05	12 00F06	BNEQ	121\$	1685
			C7	9E 00F08	MOVAB	TBIOB, IOB_PTR	1688
			53	D5 00F0D	TSTL	IOB_PTR	1694
			0C	13 00F0F	BEQL	122\$	1732
			01	DD 00F11	PUSHL	#1	1736
00000000V	EF		01	FB 00F13	CALLS	#1, VFYCAP	
	03		50	E8 00F1A	BLBS	R0, 123\$	
			F115	31 00F1D	BRW	1\$	
	08	AE 00000000G	EF	D0 00F20	MOVL	MAJOR, RECORD_HEADER	1680
	0C	AE 00000000G	EF	D0 00F28	MOVL	MINOR, RECORD_HEADER+4	1681
	10		52	D0 00F30	MOVL	LEN, RECORD_HEADER+8	1682
	6E		0C	B0 00F34	MOVW	#12, \$IOB\$OUTPUT	1683
	02	AE	02	90 00F37	MOVB	#2, \$IOB\$OUTPUT+2	
	03	AE	01	90 00F3B	MOVB	#1, \$IOB\$OUTPUT+3	
	04	AE	AE	9E 00F3F	MOVAB	RECORD_HEADER, \$IOB\$OUTPUT+4	
	44	A3	6E	9E 00F44	MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
	2C	A3	07	90 00F48	MOVB	#7, 44(IOB_PTR)	
			5B	DD 00F4C	PUSHL	R11	
			7E	D4 00F4E	CLRL	-(SP)	
			53	DD 00F50	PUSHL	IOB_PTR	
00000000G	EF		03	FB 00F52	CALLS	#3, XPO\$PUT	1685
			52	D5 00F59	TSTL	LEN	1688
			C0	13 00F5B	BEQL	122\$	
	52		03	C0 00F5D	ADDL2	#3, R2	
	52		04	C6 00F60	DIVL2	#4, R2	
6E	52		04	A5 00F63	MULW3	#4, R2, \$IOB\$OUTPUT	
	02	AE	02	90 00F67	MOVB	#2, \$IOB\$OUTPUT+2	
	03	AE	01	90 00F6B	MOVB	#1, \$IOB\$OUTPUT+3	
	04	AE	A7	D0 00F6F	MOVL	TOCIOB+56, \$IOB\$OUTPUT+4	
	44	A3	6E	9E 00F74	MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
	2C	A3	07	90 00F78	MOVB	#7, 44(IOB_PTR)	
			5B	DD 00F7C	PUSHL	R11	
			7E	D4 00F7E	CLRL	-(SP)	
			53	DD 00F80	PUSHL	IOB_PTR	
			0121	31 00F82	BRW	132\$	
	03		54	D1 00F85	CMPL	R4, #3	1694
			03	13 00F88	BEQL	125\$	
			0122	31 00F8A	BRW	133\$	
		03B4	C7	D4 00F8D	CLRL	ERRCNT	1732
	67		04	B0 00F91	MOVW	#4, IOB\$+52	1736
	02	A7	02	90 00F94	MOVB	#2, IOB\$+54	
	F8	A7	06	90 00F98	MOVB	#6, IOB\$+44	

			5B DD 00F9C	PUSHL R11	
			7E D4 00F9E	CLRL -(SP)	
		CC	A7 9F 00FA0	PUSHAB IOB\$	
00000000G	EF		03 FB 00FA3	CALLS #3, XPOSGET	
00000000G	EF	04	B7 D0 00FAA	MOVL @TOCIOB+56, TXTLEN	1737
	50	00000000G	EF D0 00FB2	MOVL TXTLEN, R0	1741
			4B 15 00FB9	BLEQ 126\$	
	50		03 C0 00FBB	ADDL2 #3, R0	1751
	50		04 C6 00FBE	DIVL2 #4, R0	
67	50		04 A5 00FC1	MULW3 #4, R0, IOB\$+52	
	02	A7	02 90 00FC5	MOVB #2, IOB\$+54	
	F8	A7	06 90 00FC9	MOVB #6, IOB\$+44	
			5B DD 00FCD	PUSHL R11	
			7E D4 00FCF	CLRL -(SP)	
		CC	A7 9F 00FD1	PUSHAB IOB\$	
00000000G	EF		03 FB 00FD4	CALLS #3, XPOSGET	
08	AE		03 D0 00FDB	MOVL #3, RECORD_HEADER	1752
	50	00000000G	EF D0 00FDF	MOVL MINOR, R0	1753
	0C	AE	50 D0 00FE6	MOVL R0, RECORD_HEADER+4	
10	AE	00000000G	EF D0 00FEA	MOVL TXTLEN, RECORD_HEADER+8	1754
			50 D5 00FF2	TSTL R0	1756
		04	13 12 00FF4	BNEQ 127\$	
		00000000G	A7 DD 00FF6	PUSHL TOCIOB+56	1762
00000000G	EF		EF DD 00FF9	PUSHL TXTLEN	
			02 FB 00FFF	CALLS #2, INSRNT	
		F02C	31 01006	BRW 1\$	1756
			03 18 01009	BGEQ 128\$	1769
	50		50 CE 0100B	MNEGL R0, R0	
	03		50 D1 0100E	CMPL R0, #3	
			0D 14 01011	BGTR 129\$	
	7E		01 CE 01013	MNEGL #1, -(SP)	
00000000V	EF		01 FB 01016	CALLS #1, VFYCAP	
	E6		50 E9 0101D	BLBC R0, 126\$	
	50	00000000G	EF D0 01020	MOVL MINOR, R0	1777
			03 18 01027	BGEQ 130\$	
	50		50 CE 01029	MNEGL R0, R0	
00000000G	EF		50 D1 0102C	CMPL R0, NXTSTC	
			07 15 01033	BLEQ 131\$	
00000000G	50	00000000G	EF D0 01035	MOVL NXTSTC, R0	
	EF		50 D0 0103C	MOVL R0, NXTSTC	
	6E		0C B0 01043	MOVW #12, \$IOB\$OUTPUT	1778
	02	AE	02 90 01046	MOVB #2, \$IOB\$OUTPUT+2	
	03	AE	01 90 0104A	MOVB #1, \$IOB\$OUTPUT+3	
	04	AE	AE 9E 0104E	MOVAB RECORD_HEADER, \$IOB\$OUTPUT+4	
00000000G	EF	08	6E 9E 01053	MOVAB \$IOB\$OUTPUT, IOB\$+68	
00000000G	EF		07 90 0105A	MOVB #7, IOB\$+44	
			5B DD 01061	PUSHL R11	
			7E D4 01063	CLRL -(SP)	
		00000000G	EF 9F 01065	PUSHAB IOB\$	
00000000G	EF		03 FB 0106B	CALLS #3, XPOSPUT	
50	EF		03 C1 01072	ADDL3 #3, TXTLEN, R0	1780
	50		04 C6 0107A	DIVL2 #4, R0	
6E	50		04 A5 0107D	MULW3 #4, R0, \$IOB\$OUTPUT	
	02	AE	02 90 01081	MOVB #2, \$IOB\$OUTPUT+2	
	03	AE	01 90 01085	MOVB #1, \$IOB\$OUTPUT+3	
	04	AE	A7 D0 01089	MOVL TOCIOB+56, \$IOB\$OUTPUT+4	
00000000G	EF	04	6E 9E 0108E	MOVAB \$IOB\$OUTPUT, IOB\$+68	

00000000G	EF	07	90	01095	MOVB	#7, IOB\$+44	:
		5B	DD	0109C	PUSHL	R11	:
		7E	D4	0109E	CLRL	-(SP)	:
00000000G	EF	00000000G	EF	9F	010A0	PUSHAB	IOB\$
			03	FB	010A6	132\$:	CALLS
			1D	11	010AD	BRB	135\$
	03B4		C7	D6	010AF	133\$:	INCL
	03B4		C7	D1	010B3		CMPL
			12	14	010B8		BGTR
	E8		A7	9F	010BA		PUSHAB
			01	DD	010BD		PUSHL
			8F	DD	010BF		PUSHL
00000000G	00	00000000G	03	FB	010C5	134\$:	CALLS
			EF	66	31	010CC	135\$:
12	6A		05	E1	010CF	136\$:	BRW
	E8		A7	9F	010D3		BBC
			01	DD	010D6		PUSHAB
			8F	DD	010D8		PUSHL
00000000G	00	00000000G	03	FB	010DE		PUSHL
F8	A7		02	90	010E5	137\$:	CALLS
			5B	DD	010E9		MOVB
			7E	D4	010EB		PUSHL
			A7	9F	010ED		CLRL
00000000G	EF	CC	03	FB	010F0		PUSHAB
50			01	DD	010F7		IOB\$
			04	010FA		CALLS	#3, XPOS\$CLOSE
			50	D4	010FB	138\$:	MOVL
			04	010FD		RET	#1, R0
						CLRL	R0
						RET	

; Routine Size: 4350 bytes, Routine Base: \$CODE\$ + 0000

```
1196 1828 1 %SBTTL 'TOCINI - Initialization'
1197 1829 1 ROUTINE tocini =
1198 1830 1 ++
1199 1831 1
1200 1832 1 FUNCTIONAL DESCRIPTION:
1201 1833 1
1202 1834 1 This routine is called by toc () to do initialization.
1203 1835 1
1204 1836 1 It opens the input file and determines the input file type (BRN,
1205 1837 1 BTC or invalid).
1206 1838 1
1207 1839 1 The output file, and several temporary files used to store
1208 1840 1 figure, table and example information are opened if they are not
1209 1841 1 already open. If they are not already open, the command line
1210 1842 1 text is prettied up for output to the output file and the output
1211 1843 1 file prologue is written.
1212 1844 1
1213 1845 1 FORMAL PARAMETERS:
1214 1846 1
1215 1847 1 None
1216 1848 1
1217 1849 1 IMPLICIT INPUTS:
1218 1850 1
1219 1851 1 CMDBLK - Command line information block
1220 1852 1
1221 1853 1 IMPLICIT OUTPUTS:
1222 1854 1
1223 1855 1 CMDBLK - Command line is reformatted
1224 1856 1 FILE TYPE - Set to BRN_FILE if .BRN or MAJ_NEW_TOC if .BTC
1225 1857 1 CAPTYP - Set to zero if output files not opened
1226 1858 1 CAPMSG - Set to zero if output files not opened
1227 1859 1 NXTSTC - Set to INFINITY if output files not opened
1228 1860 1 CHROUT - Set to zero if output files not opened
1229 1861 1 LASTHL - Set to zero if output files not opened
1230 1862 1 TOCIOB - Input file IOB
1231 1863 1 TOCOOB - Output file IOB
1232 1864 1 STCIOB - Send TOC temp file IOB
1233 1865 1 FGIOB - Header Figure temp file IOB
1234 1866 1 TBIOB - Header Table temp file IOB
1235 1867 1 EXIOB - Header Example temp file IOB
1236 1868 1 LP - CHSPTR (LINE)
1237 1869 1 INTLIN - Set to zero
1238 1870 1 EXTLIN - Set to zero
1239 1871 1
1240 1872 1 ROUTINE VALUE:
1241 1873 1 COMPLETION CODES:
1242 1874 1
1243 1875 1 None
1244 1876 1
1245 1877 1 SIDE EFFECTS:
1246 1878 1
1247 1879 1 None
1248 1880 1 --
1249 1881 2 BEGIN
1250 1882 2
1251 1883 2 LOCAL
1252 1884 2 parse_block : $xpo_spec_block;
```

```

: 1253      1885      2
: 1254      1886      2
: 1255      1887      2      | Initialization
: 1256      1888      2
: 1257      1889      2      | clr_line ();
: 1258      1890      2
: 1259      1891      2      | IF .cndblk [contents$y_tms11] THEN cndblk [contents$g_page_width] = tms_line_length;
: 1260      1892      2
: 1261      1893      2      | $xpo_iob_init (iob = tociob);
: 1262      1894      2
: 1263      P 1895      2      | IF NOT $xpo_parse_spec (spec_block = parse_block, failure = 0,
: 1264      1896      2      |     file_spec = cndblk [contents$t_input_file])
: 1265      1897      2      | THEN
: 1266      1898      2      |     | Parse failed. Set parse_block [xpo$h_file_type] non-zero to force
: 1267      1899      2      |     | quick open failure message.
: 1268      1900      2      |     |
: 1269      1901      2      |     | parse_block [xpo$h_file_type] = 1;
: 1270      1902      2      |
: 1271      1903      2      | IF .parse_block [xpo$h_file_type] NEQ 0
: 1272      1904      2      | THEN
: 1273      1905      2      | BEGIN
: 1274      1906      2      |     | User specified a file type. Open file with no defaults.
: 1275      1907      2      |     |
: 1276      1908      2      |     | $xpo_open (iob = tociob, file_spec = cndblk [contents$t_input_file], attributes = BINARY
: 1277      1909      2      |     |     %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
: 1278      P 1910      2      |     | );
: 1279      P 1911      2      |     | END
: 1280      1912      2      | ELSE
: 1281      1913      2      | BEGIN
: 1282      1914      2      |     | User did not specify a file type. Try .BRN first.
: 1283      1915      2      |     |
: 1284      1916      2      |     | IF $xpo_open (iob = tociob, file_spec = cndblk [contents$t_input_file], default = ('.BRN'),
: 1285      1917      2      |     |     attributes = BINARY, failure = 0) NEQ xpo$normal
: 1286      1918      2      |     | THEN
: 1287      P 1919      2      |     | BEGIN
: 1288      1920      2      |     |     | No .BRN file. Try .BTC
: 1289      1921      2      |     |     |
: 1290      1922      2      |     |     | $xpo_open (iob = tociob, file_spec = cndblk [contents$t_input_file], default = '.BTC',
: 1291      1923      2      |     |     |     attributes = BINARY
: 1292      1924      2      |     |     |     %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
: 1293      1925      2      |     |     | );
: 1294      P 1926      2      |     |     | END;
: 1295      P 1927      2      |     | END;
: 1296      P 1928      2      |     |
: 1297      1929      2      |     | IF .cndblk [contents$y_log]
: 1298      1930      2      |     | THEN
: 1299      1931      2      |     |     |
: 1300      1932      2      |     |     | %IF %BLISS (BLISS32)
: 1301      1933      2      |     |     | %THEN
: 1302      1934      2      |     |     | SIGNAL (contents$procfile, 1, tociob [iob$t_resultant]);
: 1303      1935      2      |     |     |
: 1304      L 1936      2      |     |     | %ELSE
: 1305      1937      2      |     |     | $xpo_put_msg (severity = success,
: 1306      1938      2      |     |     | string = $str_concat ('processing file ', tociob [iob$t_resultant], ''));
: 1307      U 1939      2      |
: 1308      U 1940      2      |
: 1309      U 1941      2

```

```
1310 1942 2 %FI
1311 1943 2
1312 1944 2 IF NOT .tocoob [iob$v_open]
1313 1945 2 THEN
1314 1946 2 BEGIN
1315 1947 2
1316 1948 2     Output file not open.
1317 1949 2     Open it and do other once only initialization.
1318 1950 2
1319 1951 2 LOCAL
1320 1952 2     file_type,
1321 1953 2     cmd_line : REF $str_descriptor (),
1322 1954 2     cmd_len,
1323 1955 2     cmd_ptr,
1324 1956 2     ptr;
1325 1957 2
1326 1958 2     |
1327 1959 2     | General initialization
1328 1960 2     |
1329 1961 2     captyp = 0;                ! No captions seen yet
1330 1962 2     capmsg = false;          ! No message written
1331 1963 2     nxtstc = infinity;        ! No .SEND TOC n's seen yet
1332 1964 2     chrout = 0;              ! No characters written to output
1333 1965 2     lasthl = 0;              ! Previous HL level
1334 1966 2
1335 1967 2 IF .cmdblk [contents$v_tms11]
1336 1968 2 THEN
1337 1969 2 BEGIN
1338 1970 2     hl_n = 0;                  ! Current HL level
1339 1971 2     file_type = CH$PTR (UPLIT ('.TMS')); ! Output file type
1340 1972 2 END
1341 1973 2 ELSE
1342 1974 2 BEGIN
1343 1975 2     hl_n = -1;                ! Current HL level
1344 1976 2     file_type = CH$PTR (UPLIT ('.RNT')); ! Output file type
1345 1977 2 END;
1346 1978 2
1347 1979 2     |
1348 1980 2     | Open output file and save resultant file name
1349 1981 2     |
1350 1982 2     $xpo_iob_init (iob = tocoob);
1351 1983 2     $xpo_open (iob = tocoob, file_spec = cmdblk [contents$t_output_file], default = (4, .file_type),
P 1984 2     related = tociob [iob$t_resultant], options = output
P 1985 2     %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
1354 1986 2 );
1355 1987 2
1356 1988 2     | Reformat command line before outputting.
1357 1989 2     |
1358 1990 2     cmd_line = cmdblk [contents$t_command_line];
1359 1991 2     cmd_len = .cmd_line [str$h_length];
1360 1992 2     cmd_ptr = .cmd_line [str$a_pointer];
1361 1993 2
1362 1994 2 WHILE NOT CH$FAIL (ptr = CH$FIND_CH (.cmd_len, .cmd_ptr, %C'<')) DO
1363 1995 2     CH$WCHAR (%C'[', .ptr);          ! Change '<' to '['
1364 1996 2
1365 1997 2 WHILE NOT CH$FAIL (ptr = CH$FIND_CH (.cmd_len, .cmd_ptr, %C'>')) DO
1366 1998 2     CH$WCHAR (%C']', .ptr);          ! Change '>' to ']'
```



```
1367 1999 3
1368 2000 3      WHILE NOT CH$FAIL (ptr = CH$FIND_CH (.cmd_len, .cmd_ptr, '%C';')) DO
1369 2001 3          CH$WLHAR (%C'.', .ptr);
1370 2002 3          ! Change ';' to '.'
1371 2003 3      IF .cmdblk [contents$v_tms11]
1372 2004 3      THEN
1373 2005 4          BEGIN
1374 2006 4              |
1375 2007 4              | Write the TMS prologue
1376 2008 4              |
1377 2009 4      %IF DSRPLUS
1378 2010 4      %THEN
1379 2011 4          put ($str_concat ('< CONTENTS version ', (.cntvrl, .cntvrp), ' >'));
1380 2012 4
1381 2013 4
1382 2014 4      %ELSE
1383 2015 4
1384 2016 4          put ($str_concat ('< DSRTOC version ', (.cntvrl, .cntvrp), ' >'));
1385 2017 4
1386 2018 4      %FI
1387 2019 4
1388 2020 4          put ($str_concat ('< ', cmdblk [contents$t_command_line], ' >'));
1389 2021 4          put ('*start*');
1390 2022 4          put ('*cinit*');
1391 2023 4          END
1392 2024 3      ELSE
1393 2025 4          BEGIN
1394 2026 4              |
1395 2027 4              | Write RUNOFF prologue
1396 2028 4              |
1397 2029 4      %IF DSRPLUS
1398 2030 4      %THEN
1399 2031 4
1400 2032 4          put ($str_concat ('.! CONTENTS version ', (.cntvrl, .cntvrp)));
1401 2033 4
1402 2034 4      %ELSE
1403 2035 4
1404 2036 4          put ($str_concat ('.! DSRTOC version ', (.cntvrl, .cntvrp)));
1405 2037 4
1406 2038 4      %FI
1407 2039 4
1408 2040 4          put ($str_concat ('.! ', cmdblk [contents$t_command_line]));
1409 2041 4          put ('.SAVE');
1410 2042 4          put ('.NO FLAGS ALL');
1411 2043 4          put ('.NO FLAGS BREAK          .NO FLAGS CAPITALIZE          .NO FLAGS ENDFOOTNOTE');
1412 2044 4          put ('.NO FLAGS HYPHENATE      .NO FLAGS INDEX          .NO FLAGS PERIOD');
1413 2045 4          put ('.NO FLAGS SPACE          .NO FLAGS SUBSTITUTE');
1414 2046 4          put ('.FLAGS ACCEPT          .FLAGS BOLD *          .FLAGS COMMENT !');
1415 2047 4          put ('.FLAGS LOWERCASE \      .FLAGS OVERSTRIKE %          .FLAGS UNDERLINE &');
1416 2048 4          put ('.FLAGS UPPERCASE ^');
1417 2049 4          put ('.FLAGS ALL');
1418 2050 4          put ('.NO FILL          .NO JUSTIFY');
1419 2051 4          put (
1420 2052 4              $str_concat (
1421 2053 4                  '.LEFT MARGIN ',
1422 2054 4                  $str_ascii (.indents [0]),
1423 2055 4                  '.RIGHT MARGIN ',
```

```
1424 P 2056 4 $str_ascii (.cndblk [contents$g_page_width]),
1425 P 2057 4 'PAGE SIZE
1426 P 2058 4 $str_ascii (.cndblk [contents$g_page_width])
1427 P 2059 4 )
1428 2060 4 );
1429 2061 4
1430 2062 4 END;
1431 2063 4
1432 2064 4
1433 2065 4 | Open SEND TOC temp file
1434 2066 4
1435 2067 4 $xpo_iob_init (iob = stciob);
1436 P 2068 4 $xpo_open (iob = stciob, file_spec = 'OCNTST.TMP', attributes = BINARY, options = output
1437 P 2069 4 %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
1438 2070 4 );
1439 2071 4
1440 2072 4 | Open figures temp file
1441 2073 4
1442 2074 4 $xpo_iob_init (iob = fgiob);
1443 P 2075 4 $xpo_open (iob = fgiob, file_spec = 'OCNTFG.TMP', attributes = BINARY, options = output
1444 P 2076 4 %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
1445 2077 4 );
1446 2078 4
1447 2079 4 | Open tables temp file
1448 2080 4
1449 2081 4 $xpo_iob_init (iob = tbiob);
1450 P 2082 4 $xpo_open (iob = tbiob, file_spec = 'OCNTTB.TMP', attributes = BINARY, options = output
1451 P 2083 4 %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
1452 2084 4 );
1453 2085 4
1454 2086 4 | Open examples temp file
1455 2087 4
1456 2088 4 $xpo_iob_init (iob = exiob);
1457 P 2089 4 $xpo_open (iob = exiob, file_spec = 'OCNTEX.TMP', attributes = BINARY, options = output
1458 P 2090 4 %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
1459 2091 4 );
1460 2092 4
1461 2093 4 IF .cndblk [contents$v_require]
1462 2094 4 THEN
1463 2095 4 BEGIN
1464 2096 4 | Include contents of require file now
1465 2097 4
1466 2098 4 LOCAL
1467 2099 4 require_iob : $xpo_iob ();
1468 2100 4
1469 2101 4
1470 2102 4 $xpo_iob_init (iob = require_iob);
1471 P 2103 4 $xpo_open (iob = require_iob,
1472 P 2104 4 file_spec = cndblk [contents$t_require_file]
1473 P 2105 4 %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
1474 2106 4 );
1475 2107 4
1476 2108 4 WHILE $xpo_get (iob = require_iob) EQL xpo$_normal DO
1477 2109 4 put (require_iob [iob$t_sstring]);
1478 2110 4
1479 2111 4 $xpo_close (iob = require_iob);
1480 2112 4 END
```

```

: 1481      2113 3      ELSE
: 1482      2114 4      BEGIN
: 1483      2115 4      :
: 1484      2116 4      :   No require file - generate standard heading
: 1485      2117 4      :
: 1486      2118 4      IF NOT .cmdblk [contents$y_tms11]
: 1487      2119 4      THEN
: 1488      2120 5      BEGIN
: 1489      2121 5      put ('.CENTER;CONTENTS');
: 1490      2122 5      put ('.BLANK');
: 1491      2123 4      END;
: 1492      2124 4      :
: 1493      2125 3      END;
: 1494      2126 3      :
: 1495      2127 2      END;
: 1496      2128 2      :
: 1497      2129 2      :
: 1498      2130 2      :   Read file information record
: 1499      2131 2      :
: 1500      2132 2      IF $xpo_get (iob = tociob, fullwords = 2) NEQ xpo$_normal
: 1501      2133 2      THEN
: 1502      2134 3      BEGIN
: 1503      2135 3      :
: 1504      2136 3      :   Empty input file
: 1505      2137 3      :
: 1506      L 2133 3      %IF %BLISS (BLISS32)
: 1507      2139 3      %THEN
: 1508      2140 3      SIGNAL (contents$_emptyin, 1, tociob [iob$t_resultant]);
: 1509      U 2141 3      %ELSE
: 1510      2142 3      $xpo_put_msg (severity = warning,
: 1511      U 2143 3      string = $str_concat ('empty input file ',
: 1512      U 2144 3      tociob [iob$t_resultant], ''));
: 1513      2145 3      %FI
: 1514      2146 3      :
: 1515      2147 3      RETURN false;
: 1516      2148 3      END
: 1517      2149 2      ELSE
: 1518      2150 3      BEGIN
: 1519      2151 3      :
: 1520      2152 3      :   Determine file type
: 1521      2153 3      :
: 1522      2154 3      BIND
: 1523      2155 3      codes = .toclob [iob$a_data] : VECTOR;
: 1524      2156 3      :
: 1525      2157 3      LOCAL
: 1526      2158 3      correct_format;
: 1527      2159 3      :
: 1528      2160 3      correct_format = true;      ! Assume file format is correct
: 1529      2161 3      :
: 1530      2162 3      SELECT ONE .codes [0] OF
: 1531      2163 3      SET
: 1532      2164 3      :
: 1533      2165 3      [brn_file] :
: 1534      2166 4      BEGIN
: 1535      2167 4      :
: 1536      2168 4      :   Validate .BRN format
: 1537      2169 4      :

```

```

: 1538      2170  4      file_type = brn_file;
: 1539      2171  4
: 1540      2172  4      !F .codes [1] NEQ brn_ident THEN correct_format = false;
: 1541      2173  3      END;
: 1542      2174  3
: 1543      2175  3      [maj_new_toc] :
: 1544      2176  4      BEGIN
: 1545      2177  4      |
: 1546      2178  4      | Validate .BTC format
: 1547      2179  4      |
: 1548      2180  4      | file_type = maj_new_toc;
: 1549      2181  4      |
: 1550      2182  4      | IF .codes [1] NEQ toc_format THEN correct_format = false;
: 1551      2183  3      | END;
: 1552      2184  3
: 1553      2185  3      [OTHERWISE] :
: 1554      2186  3      |
: 1555      2187  3      | Invalid file format
: 1556      2188  3      |
: 1557      2189  3      | correct_format = false;
: 1558      2190  3
: 1559      2191  3      TES;
: 1560      2192  3
: 1561      2193  3      IF NOT .correct_format
: 1562      2194  3      THEN
: 1563      2195  4      BEGIN
: 1564      2196  4      |
: 1565      2197  4      | Report bad file format and quit
: 1566      2198  4      |
: 1567      2199  4      | %IF %BLISS (BLISS32)
: 1568      2200  4      | %THEN
: 1569      2201  4      |     SIGNAL (contents$_invinput, 1, toclob [iob$t_resultant]);
: 1570      2202  4      | %ELSE
: 1571      2203  4      |     $xpo_put_msg (severity = warning,
: 1572      2204  4      |         string = $str_concat ('invalid file format ',
: 1573      2205  4      |             toclob [iob$t_resultant], ''));
: 1574      2206  4      | %FI
: 1575      2207  4      |
: 1576      2208  4      | RETURN false;
: 1577      2209  3      | END;
: 1578      2210  3
: 1579      2211  2      END;
: 1580      2212  2
: 1581      2213  2      RETURN true;
: 1582      2214  1      END;

```

										.PSECT		\$PLITS,NOWRT,NOEXE,2							
										4E	52	42	2E	0008A	P.ABA:	.ASCII	\\.BRN\\	:	
										43	54	42	2E	0008E	P.ABB:	.ASCII	\\.BTC\\	:	
														00092		.BLKB	2	:	
										53	4D	54	2E	00094	P.ABC:	.ASCII	\\.TMS\\	:	
										54	4E	52	2E	00098	P.ABD:	.ASCII	\\.RNT\\	:	
6F	69	73	72	65	76	20	43	4F	54	52	53	44	20	3C	0009C	P.ABG:	.ASCII	\\< DSRTOC version \\	:
														20	6E	000AB			:

69	73	72	65	76	20	43	4F	2A	74	72	61	74	73	2A	000AD	P.ABH:	.ASCII	\>\	
								2A	74	69	6E	69	63	2A	000AF	P.ABM:	.ASCII	\<\	
								54	52	53	44	20	21	2E	000B1	P.ABN:	.ASCII	\>\	
															000B3	P.ABQ:	.ASCII	*start*\	
															000BA	P.ABR:	.ASCII	*cinit*\	
															000C1	P.ABT:	.ASCII	\.! DSRTOC version \	
															000D0				
															000D3	P.ABW:	.ASCII	\.! \	
4B	41	4C	4C	41	20	53	47	41	4C	46	20	4F	4E	2E	000D6	P.ABY:	.ASCII	\.SAVE\	
4C	46	20	4F	4E	2E	09	45	5A	49	4C	41	54	49	50	000DB	P.ABZ:	.ASCII	\.NO FLAGS ALL\	
45	54	4F	4E	54	4F	4F	46	44	4E	45	20	53	47	41	000E8	P.ACA:	.ASCII	\.NO FLAGS BREAK\<9><9>\.NO FLAGS CA\	
45	48	50	59	48	20	53	47	41	4C	46	20	4F	4E	2E	00105		.ASCII	\PITALIZE\<9>\.NO FLAGS ENDFOOTNOTE\	
20	53	47	41	4C	46	20	4F	4E	2E	09	45	54	41	4E	00114				
															00123	P.ACB:	.ASCII	\.NO FLAGS HYPHENATE\<9>\.NO FLAGS INDE\	
															00141				
45	50	20	53	47	41	4C	46	20	4F	4E	2E	09	09	58	00145		.ASCII	\X\<9><9>\.NO .LAGS PERIOD\	
															00154				
45	43	41	50	53	20	53	47	41	4C	46	20	4F	4E	2E	00158	P.ACC:	.ASCII	\.NO FLAGS SPACE\<9><9>\.NO FLAGS SU\	
	55	53	20	53	47	41	4C	46	20	4F	4E	2E	09	09	00167				
5F	20	54	50	45	43	43	41	20	53	47	41	4C	46	2E	00175		.ASCII	\BSTITUTE\	
	20	44	4C	4F	42	20	53	47	41	4C	46	2E	09	09	0017D	P.ACD:	.ASCII	\.FLAGS ACCEPT _\<9><9>\.FLAGS BOLD \	
45	4D	4D	4F	43	20	53	47	41	4C	46	2E	09	09	2A	0018C		.ASCII	*\<9><9>\.FLAGS COMMENT !\	
															0019A				
53	41	43	52	45	57	4F	4C	20	53	47	41	4C	46	2E	001A9		.ASCII	\.FLAGS LOWERCASE \<92><9>\.FLAGS OVE\	
	45	56	4F	20	53	47	41	4C	46	2E	09	5C	20	45	001AD	P.ACE:	.ASCII	\.FLAGS LOWERCASE \<92><9>\.FLAGS OVE\	
47	41	4C	46	2E	09	25	20	45	48	49	52	54	53	52	001BC		.ASCII	\RSTRIKE X\<9>\.FLAGS UNDERLINE &\	
		26	20	45	49	4C	52	45	44	4E	55	20	53	001CA					
53	41	43	52	45	50	50	55	20	53	47	41	4C	46	2E	001D9		.ASCII	\.FLAGS UPPERCASE ^\	
															001E6	P.ACF:	.ASCII	\.FLAGS UPPERCASE ^\	
															001F5				
4A	20	4F	4E	2E	09	09	4C	4C	49	46	20	4F	4E	2E	001F8	P.ACG:	.ASCII	\.FLAGS ALL\	
															00202	P.ACH:	.ASCII	\.NO FILL\<9><9>\.NO JUSTIFY\	
															00211				
4E	49	20	4E	49	47	52	41	4D	20	54	46	45	4C	2E	00217	P.ACL:	.ASCII	\.LEFT MARGIN \	
															00224	P.ACM:	.ASCII	<9><9>\.RIGHT MARGIN \	
															00233				
		20	2C	20	45	5A	49	53	20	45	47	41	50	2E	00234	P.ACN:	.ASCII	<9>\.PAGE SIZE , \	
															00242	P.ACR:	.ASCII	\OCNTST.TMP\	
															0024C	P.ACS:	.ASCII	\OCNTFG.TMP\	
															00256	P.ACT:	.ASCII	\OCNTTB.TMP\	
															00260	P.ACU:	.ASCII	\OCNTEX.TMP\	
54	4E	45	54	4E	4F	43	3B	52	45	54	4E	45	43	2E	0026A	P.ACV:	.ASCII	\.CENTER;CONTENTS\	
															00279				
															0027A	P.ACW:	.ASCII	\.BLANK\	

.PSECT \$OWNS,NOEXE,2

```

0004 0046C $IOB$DEFAULT:
      01 0E 0046E .WORD 4
00000000' 00470 .BYTE 14, 1
      0004 00474 $IOB$DEFAULT:
      01 0E 00476 .WORD 4
00000000' 00478 .BYTE 14, 1
           .ADDRESS P.ABA
           .ADDRESS P.ABB

```

```

0011 0047C $STR$STRING0:
      01 0E 0047E .WORD 17
00000000' 00480 .BYTE 14, 1
      0002 00484 $STR$STRING2:
      01 0E 00486 .WORD 2
00000000' 00488 .BYTE 14, 1
      0002 0048C $STR$STRING0:
      01 0E 0048E .WORD 2
00000000' 00490 .BYTE 14, 1
      0002 00494 $STR$STRING2:
      01 0E 00496 .WORD 2
00000000' 00498 .BYTE 14, 1
      0007 0049C $STR$STRING:
      01 0E 0049E .WORD 7
00000000' 004A0 .BYTE 14, 1
      0007 004A4 $STR$STRING:
      01 0E 004A6 .WORD 7
00000000' 004A8 .BYTE 14, 1
      0012 004AC $STR$STRING0:
      01 0E 004AE .WORD 18
00000000' 004B0 .BYTE 14, 1
      0003 004B4 $STR$STRING0:
      01 0E 004B6 .WORD 3
00000000' 004B8 .BYTE 14, 1
      0005 004BC $STR$STRING:
      01 0E 004BE .WORD 5
00000000' 004C0 .BYTE 14, 1
      000D 004C4 $STR$STRING:
      01 0E 004C6 .WORD 13
00000000' 004C8 .BYTE 14, 1
      003B 004CC $STR$STRING:
      01 0E 004CE .WORD 59
00000000' 004D0 .BYTE 14, 1
      0035 004D4 $STR$STRING:
      01 0E 004D6 .WORD 53
00000000' 004D8 .BYTE 14, 1
      0025 004DC $STR$STRING:
      01 0E 004DE .WORD 37
00000000' 004E0 .BYTE 14, 1
      0030 004E4 $STR$STRING:
      01 0E 004E6 .WORD 48
00000000' 004E8 .BYTE 14, 1
      0039 004EC $STR$STRING:

```

.....

.....

```

01 0E 004EE .WORD 57
00000000' 004F0 .BYTE 14, 1
0012 004F4 $STR$STRING: .ADDRESS P.ACE
                                .WORD 18
01 0E 004F6 .BYTE 14, 1
00000000' 004F8 .ADDRESS P.ACF
000A 004FC $STR$STRING:
                                .WORD 10
01 0E 004FE .BYTE 14, 1
00000000' 00500 .ADDRESS P.ACG
0015 00504 $STR$STRING:
                                .WORD 21
01 0E 00506 .BYTE 14, 1
00000000' 00508 .ADDRESS P.ACH
000D 0050C $STR$STRING:
                                .WORD 13
01 0E 0050E .BYTE 14, 1
00000000' 00510 .ADDRESS P.ACL
0010 00514 $STR$STRING2:
                                .WORD 16
01 0E 00516 .BYTE 14, 1
00000000' 00518 .ADDRESS P.ACM
000E 0051C $STR$STRING4:
                                .WORD 14
01 0E 0051E .BYTE 14, 1
00000000' 00520 .ADDRESS P.ACN
000A 00524 $IOB$FILE SPEC:
                                .WORD 10
01 0E 00526 .BYTE 14, 1
00000000' 00528 .ADDRESS P.ACR
000A 0052C $IOB$FILE SPEC:
                                .WORD 10
01 0E 0052E .BYTE 14, 1
00000000' 00530 .ADDRESS P.ACS
000A 00534 $IOB$FILE SPEC:
                                .WORD 10
01 0E 00536 .BYTE 14, 1
00000000' 00538 .ADDRESS P.ACT
000A 0053C $IOB$FILE SPEC:
                                .WORD 10
01 0E 0053E .BYTE 14, 1
00000000' 00540 .ADDRESS P.ACU
0010 00544 $STR$STRING:
                                .WORD 16
01 0E 00546 .BYTE 14, 1
00000000' 00548 .ADDRESS P.ACV
0006 0054C $STR$STRING:
                                .WORD 6
01 0E 0054E .BYTE 14, 1
00000000' 00550 .ADDRESS P.ACW

```

```

IOB$= TOC10B
IOB$RESULTANT= TOC10B+28
IOB$= TOC10B
IOB$= TOC10B
IOB$= TOC10B

```

\$IOB\$RELATED= TOCIOB+28
IOB\$= FGIOB
IOB\$RESULTANT= FGIOB+28
IOB\$= FGIOB
IOB\$= TBIOB
IOB\$RESULTANT= TBIOB+28
IOB\$= TBIOB
IOB\$= EXIOB
IOB\$RESULTANT= EXIOB+28
IOB\$= EXIOB
IOB\$= TOCIOB
.EXTRN XPOS\$PARSE_SPEC, XPOS\$OPEN
.EXTRN XST\$FORMAT
.PSECT \$CODE\$,NOWRT,2

				OFFC 00000	TOCINI:	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	1829
		5B	00000000G	EF 9E 00002		MOVAB	XPOS\$FAILURE, R11	
		5A	00000000G	EF 9E 00009		MOVAB	CMDBLK, R10	
		59	00000000G	EF 9E 00010		MOVAB	\$STR\$TARGET, R9	
		58	00000000G	EF 9E 00017		MOVAB	IOB\$, R8	
		57	00000000'	EF 9E 0001E		MOVAB	IOB\$, R7	
		5E	FEC4	CE 9E 00025		MOVAB	-316(SP), SP	
	00000000G	EF	00000000G	EF 9E 0002A		MOVAB	LINE, LP	1889
			00000000G	EF D4 00035		CLRL	INTLIN	
			00000000G	EF D4 0003B		CLRL	EXTLIN	
06		6A		01 E1 00041		BBC	#1, CMDBLK, 1\$	1891
	18	AA	0258	8F 3C 00045		MOVZWL	#600, CMDBLK+24	
00F4	8F	00	6E	00 2C 0004B	1\$:	MOVCS	#0, (SP), #0, #244, IOB\$	1893
				67 00052				
		67	0301003D	8F D0 00053		MOVL	#50397245, IOB\$	
	1E	A7	020E	8F B0 0005A		MOVW	#526, IOB\$RESULTANT+2	
				7E 7C 00060		CLRQ	-(SP)	1896
		7E		01 CE 00062		MNEGL	#1, -(SP)	
			B8	AD 9F 00065		PUSHAB	PARSE_BLOCK	
			3C	AA 9F 00068		PUSHAB	\$STR\$FILE_SPEC	
	00000000G	EF		05 FB 0006B		CALLS	#5, XPOS\$PARSE_SPEC	
		04		50 E8 00072		BLBS	R0, 2\$	
	E0	AD		01 B0 00075		MOVW	#1, PARSE_BLOCK+40	1902
	04	50	3C	AA 9E 00079	2\$:	MOVAB	\$IOB\$FILE_SPEC, R0	1912
		A7		50 D0 0007D		MOVL	R0, IOB\$+Z	
			E0	AD B5 00081		TSTW	PARSE_BLOCK+40	1904
				2D 12 00084		BNEQ	3\$	
	08	A7	046C	C7 9E 00086		MOVAB	\$IOB\$DEFAULT, IOB\$+8	1920
	30	A7		01 88 0008C		BISB2	#1, IOB\$+48	
	2C	A7		01 90 00090		MOVAB	#1, IOB\$+44	
				7E 7C 00094		CLRQ	-(SP)	
				57 DD 00096		PUSHL	R7	
	00000000G	EF		03 FB 00098		CALLS	#3, XPOS\$OPEN	
	00208001	8F		50 D1 0009F		CMPL	R0, #2129921	
				24 13 000A6		BEQL	4\$	
	04	A7	3C	AA 9E 000A8		MOVAB	\$IOB\$FILE_SPEC, IOB\$+4	1929
	08	A7	0474	C7 9E 000AD		MOVAB	\$IOB\$DEFAULT, IOB\$+8	
	30	A7		01 88 000B3	3\$:	BISB2	#1, IOB\$+48	
	2C	A7		01 90 000B7		MOVAB	#1, IOB\$+44	
			00000000G	EF 9F 000BB		PUSHAB	OPEN_ERROR	
				7E D4 000C1		CLRL	-(SP)	

[illegible]

62	53	3B	3A	001AC	13\$:	LOCC	#59, CMD_LEN, (CMD_PTR)
		02	12	001B0		BNEQ	14\$
		51	D4	001B2		CLRL	R1
		51	D5	001B4	14\$:	TSTL	PTR
		05	13	001B6		BEQL	15\$
	61	2E	90	001B8		MOVB	#46, (PTR)
		EF	11	001BB		BRB	13\$
03	6A	01	E0	001BD	15\$:	BBS	#1, CMDBLK, 16\$
		01	0C	001C1		BRW	20\$
	B0	AD	00000000G	EF	B0	001C4	16\$:
	B2	AD		0E	90	001CC	MOVW
	B3	AD		01	90	001D0	MOVB
	B4	AD	00000000G	CF	D0	001D4	MOVL
			0484	C7	9F	001DC	PUSHAB
			B0	AD	9F	001E0	PUSHAB
			047C	C7	9F	001E3	PUSHAB
	00000000G	EF		03	FB	001E7	CALLS
			00000000G	EF	9F	001EE	PUSHAB
				7E	D4	001F4	CLRL
			0201	8F	BB	001F6	PUSHR
				7E	D4	001FA	CLRL
	00000000G	EF		05	FB	001FC	CALLS
		50		69	3C	00203	MOVZWL
	00000000G	EF		50	C0	00206	ADDL2
		44		69	9E	0020D	MOVAB
		2C		07	90	00211	MOVB
				5B	DD	00215	PUSHL
				7E	D4	00217	CLRL
				58	DD	00219	PUSHL
07	00000000G	EF		03	FB	0021B	CALLS
		6A		01	E1	00222	BBC
	00000000G	EF		00	FB	00226	CALLS
			0494	C7	9F	0022D	17\$:
			54	AA	9F	00231	PUSHAB
			048C	C7	9F	00234	PUSHAB
	00000000G	EF		03	FB	00238	PUSHAB
			00000000G	EF	9F	0023F	CALLS
				7E	D4	00245	PUSHAB
			0201	8F	BB	00247	STR\$FAILURE
				7E	D4	0024B	CLRL
	00000000G	EF		05	FB	0024D	CLRL
		50		69	3C	00254	CLRL
	00000000G	EF		50	C0	00257	CALLS
		44		69	9E	0025E	#5, XST\$COPY
		2C		07	90	00262	MOVZWL
				5B	DD	00266	TMPSTR, R0
				7E	D4	00268	R0, CHROUT
				58	DD	0026A	MOVAB
	00000000G	EF		03	FB	0026C	\$IOB\$OUTPUT, IOB\$+68
		6A		01	E1	00273	MOVAB
	00000000G	EF		00	FB	00277	MOVB
			00000000G	EF	9F	0027E	#7, IOB\$+44
				7E	D4	00284	PUSHL
			049C	59	DD	00286	R1
				C7	9F	00288	CLRL
	00000000G	EF		7E	D4	0028C	CLRL
				05	FB	0028E	CLRL
							CALLS

2000

2001

2003

2016

2020

2021

			69	3C	00295	MOVZWL	TMPSTR, R0		
	00000000G	50	50	C0	00298	ADDL2	R0, CHROUT		
	44	EF	69	9E	0029F	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
	2C	A8	07	90	002A3	MOVB	#7, IOB\$+44		
			58	DD	002A7	PUSHL	R11		
			7E	D4	002A9	CLRL	-(SP)		
			58	DD	002AB	PUSHL	R8		
07	00000000G	EF	03	FB	002AD	CALLS	#3, XPO\$PUT		
	6A		01	E1	002B4	BBC	#1, CMDBLK, 19\$		
	00000000G	EF	00	FB	002B8	CALLS	#0, SPLIT		
			EF	9F	002BF	PUSHAB	STR\$FAILURE	2022	
		00000000G	7E	D4	002C5	CLRL	-(SP)		
			59	DD	002C7	PUSHL	R9		
		04A4	C7	9F	002C9	PUSHAB	\$STR\$STRING		
			039A	31	002CD	BRW	33\$	2036	
	B0	AD	00000000G	EF	B0	002D0	CNTVRL, \$STR\$STRING1		
	B2	AD		0E	90	002D8	#14, \$STR\$STRING1+2		
	B3	AD		01	90	002DC	#1, \$STR\$STRING1+3		
	B4	AD	00000000G	EF	D0	002E0	CNTVRP, \$STR\$STRING1+4		
			B0	AD	9F	002E8	\$STR\$STRING1		
		04AC	C7	9F	002EB	PUSHAB	\$STR\$STRING0		
	00000000G	EF	02	FB	002EF	CALLS	#2, XST\$JOIN		
			EF	9F	002F6	PUSHAB	STR\$FAILURE		
		00000000G	7E	D4	002FC	CLRL	-(SP)		
			0201	8F	BB	002FE	PUSHR	#*M<R0,R9>	
			7E	D4	00302	CLRL	-(SP)		
	00000000G	EF	05	FB	00304	CALLS	#5, XST\$COPY		
	50		69	3C	0030B	MOVZWL	TMPSTR, R0		
	00000000G	EF	50	C0	0030E	ADDL2	R0, CHROUT		
	44	A8	69	9E	00315	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
	2C	A8	07	90	00319	MOVB	#7, IOB\$+44		
			58	DD	0031D	PUSHL	R11		
			7E	D4	0031F	CLRL	-(SP)		
			58	DD	00321	PUSHL	R8		
07	00000000G	EF	03	FB	00323	CALLS	#3, XPO\$PUT		
	6A		01	E1	0032A	BBC	#1, CMDBLK, 21\$		
	00000000G	EF	00	FB	0032E	CALLS	#0, SPLIT		
			AA	9F	00335	PUSHAB	\$STR\$STRING1	2040	
		54	C7	9F	00338	PUSHAB	\$STR\$STRING0		
		04B4	02	FB	0033C	CALLS	#2, XST\$JOIN		
	00000000G	EF	EF	9F	00343	PUSHAB	STR\$FAILURE		
			7E	D4	00349	CLRL	-(SP)		
			0201	8F	BB	0034B	PUSHR	#*M<R0,R9>	
			7E	D4	0034F	CLRL	-(SP)		
	00000000G	EF	05	FB	00351	CALLS	#5, XST\$COPY		
	50		69	3C	00358	MOVZWL	TMPSTR, R0		
	00000000G	EF	50	C0	0035B	ADDL2	R0, CHROUT		
	44	A8	69	9E	00362	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
	2C	A8	07	90	00366	MOVB	#7, IOB\$+44		
			58	DD	0036A	PUSHL	R11		
			7E	D4	0036C	CLRL	-(SP)		
			58	DD	0036E	PUSHL	R8		
07	00000000G	EF	03	FB	00370	CALLS	#3, XPO\$PUT		
	6A		01	E1	00377	BBC	#1, CMDBLK, 22\$		
	00000000G	EF	00	FB	0037B	CALLS	#0, SPLIT		
			EF	9F	00382	PUSHAB	STR\$FAILURE	2041	
		00000000G	7E	D4	00388	CLRL	-(SP)		

			04BC	59 DD 0038A	PUSHL R9	
				C7 9F 0038C	PUSHAB \$STR\$STRING	
				7E D4 00390	CLRL -(SP)	
	00000000G	EF		05 FB 00392	CALLS #5, XST\$COPY	
		50		69 3C 00399	MOVZWL TMPSTR, R0	
	00000000G	EF		50 C0 0039C	ADDL2 R0, CHROUT	
	44	A8		69 9E 003A3	MOVAB \$IOB\$OUTPUT, IOB\$+68	
	2C	A8		07 90 003A7	MOVB #7, IOB\$+44	
				5B DD 003AB	PUSHL R11	
				7E D4 003AD	CLRL -(SP)	
				58 DD 003AF	PUSHL R8	
07	00000000G	EF		03 FB 003B1	CALLS #3, XPO\$PUT	
		6A		01 E1 003B8	BBC #1, CMDBLK, 23\$	
	00000000G	EF	00000000G	00 FB 003BC	CALLS #0, SPLIT	
				EF 9F 003C3	PUSHAB STR\$FAILURE	2042
				7E D4 003C9	CLRL -(SP)	
				59 DD 003CB	PUSHL R9	
			04C4	C7 9F 003CD	PUSHAB \$STR\$STRING	
				7E D4 003D1	CLRL -(SP)	
	00000000G	EF		05 FB 003D3	CALLS #5, XST\$COPY	
		50		69 3C 003DA	MOVZWL TMPSTR, R0	
	00000000G	EF		50 C0 003DD	ADDL2 R0, CHROUT	
	44	A8		69 9E 003E4	MOVAB \$IOB\$OUTPUT, IOB\$+68	
	2C	A8		07 90 003E8	MOVB #7, IOB\$+44	
				5B DD 003EC	PUSHL R11	
				7E D4 003EE	CLRL -(SP)	
				58 DD 003F0	PUSHL R8	
07	00000000G	EF		03 FB 003F2	CALLS #3, XPO\$PUT	
		6A		01 E1 003F9	BBC #1, CMDBLK, 24\$	
	00000000G	EF	00000000G	00 FB 003FD	CALLS #0, SPLIT	
				EF 9F 00404	PUSHAB STR\$FAILURE	2043
				7E D4 0040A	CLRL -(SP)	
				59 DD 0040C	PUSHL R9	
			04CC	C7 9F 0040E	PUSHAB \$STR\$STRING	
				7E D4 00412	CLRL -(SP)	
	00000000G	EF		05 FB 00414	CALLS #5, XST\$COPY	
		50		69 3C 0041B	MOVZWL TMPSTR, R0	
	00000000G	EF		50 C0 0041E	ADDL2 R0, CHROUT	
	44	A8		69 9E 00425	MOVAB \$IOB\$OUTPUT, IOB\$+68	
	2C	A8		07 90 00429	MOVB #7, IOB\$+44	
				5B DD 0042D	PUSHL R11	
				7E D4 0042F	CLRL -(SP)	
				58 DD 00431	PUSHL R8	
07	00000000G	EF		03 FB 00433	CALLS #3, XPO\$PUT	
		6A		01 E1 0043A	BBC #1, CMDBLK, 25\$	
	00000000G	EF	00000000G	00 FB 0043E	CALLS #0, SPLIT	
				EF 9F 00445	PUSHAB STR\$FAILURE	2044
				7E D4 0044B	CLRL -(SP)	
				59 DD 0044D	PUSHL R9	
			04D4	C7 9F 0044F	PUSHAB \$STR\$STRING	
				7E D4 00453	CLRL -(SP)	
	00000000G	EF		05 FB 00455	CALLS #5, XST\$COPY	
		50		69 3C 0045C	MOVZWL TMPSTR, R0	
	00000000G	EF		50 C0 0045F	ADDL2 R0, CHROUT	
	44	A8		69 9E 00466	MOVAB \$IOB\$OUTPUT, IOB\$+68	
	2C	A8		07 90 0046A	MOVB #7, IOB\$+44	
				5B DD 0046E	PUSHL R11	

07	00000000G	EF		7E	D4	00470	CLRL	-(SP)	
		6A		58	DD	00472	PUSHL	R8	
	00000000G	EF		03	FB	00474	CALLS	#3, XPO\$PUT	
			00000000G	01	E1	0047B	BBC	#1, CMDBLK, 26\$	
				00	FB	0047F	CALLS	#0, SPLIT	
				EF	9F	00486	PUSHAB	STR\$FAILURE	2045
				7E	D4	0048C	CLRL	-(SP)	
			04DC	59	DD	0048E	PUSHL	R9	
				C7	9F	00490	PUSHAB	\$STR\$STRING	
	00000000G	EF		7E	D4	00494	CLRL	-(SP)	
		50		05	FB	00496	CALLS	#5, XST\$COPY	
	00000000G	EF		69	3C	0049D	MOVZWL	TMPSTR, R0	
		44		50	C0	004A0	ADDL2	R0, CHROUT	
		2C		69	9E	004A7	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
				07	90	004AB	MOVB	#7, IOB\$+44	
				58	DD	004AF	PUSHL	R11	
				7E	D4	004B1	CLRL	-(SP)	
				58	DD	004B3	PUSHL	R8	
07	0000C000G	EF		03	FB	004B5	CALLS	#3, XPO\$PUT	
		6A		01	E1	004BC	BBC	#1, CMDBLK, 27\$	
	00000000G	EF		00	FB	004C0	CALLS	#0, SPLIT	
			00000000G	EF	9F	004C7	PUSHAB	STR\$FAILURE	2046
				7E	D4	004CD	CLRL	-(SP)	
				59	DD	004CF	PUSHL	R9	
			04E4	C7	9F	004D1	PUSHAB	\$STR\$STRING	
				7E	D4	004D5	CLRL	-(SP)	
	00000000G	EF		05	FB	004D7	CALLS	#5, XST\$COPY	
		50		69	3C	004DE	MOVZWL	TMPSTR, R0	
	00000000G	EF		50	C0	004E1	ADDL2	R0, CHROUT	
		44		69	9E	004E8	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		2C		07	90	004EC	MOVB	#7, IOB\$+44	
				58	DD	004F0	PUSHL	R11	
				7E	D4	004F2	CLRL	-(SP)	
				58	DD	004F4	PUSHL	R8	
07	00000000G	EF		03	FB	004F6	CALLS	#3, XPO\$PUT	
		6A		01	E1	004FD	BBC	#1, CMDBLK, 28\$	
	00000000G	EF		00	FB	00501	CALLS	#0, SPLIT	
			00000000G	EF	9F	00508	PUSHAB	STR\$FAILURE	2047
				7E	D4	0050E	CLRL	-(SP)	
				59	DD	00510	PUSHL	R9	
			04EC	C7	9F	00512	PUSHAB	\$STR\$STRING	
				7E	D4	00516	CLRL	-(SP)	
	00000000G	EF		05	FB	00518	CALLS	#5, XST\$COPY	
		50		69	3C	0051F	MOVZWL	TMPSTR, R0	
	00000000G	EF		50	C0	00522	ADDL2	R0, CHROUT	
		44		69	9E	00529	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		2C		07	90	0052D	MOVB	#7, IOB\$+44	
				58	DD	00531	PUSHL	R11	
				7E	D4	00533	CLRL	-(SP)	
				58	DD	00535	PUSHL	R8	
07	00000000G	EF		03	FB	00537	CALLS	#3, XPO\$PUT	
		6A		01	E1	0053E	BBC	#1, CMDBLK, 29\$	
	00000000G	EF		00	FB	00542	CALLS	#0, SPLIT	
			00000000G	EF	9F	00549	PUSHAB	STR\$FAILURE	2048
				7E	D4	0054F	CLRL	-(SP)	
				59	DD	00551	PUSHL	R9	
			04F4	C7	9F	00553	PUSHAB	\$STR\$STRING	

	00000000G	EF	50	7E	D4	00557	CLRL	-(SP)	
				05	FB	00559	CALLS	#5, XST\$COPY	
	00000000G	EF	50	69	3C	00560	MOVZWL	TMPSTR, R0	
	44	A8		50	C0	00563	ADDL2	R0, CHROUT	
	2C	A8		69	9E	0056A	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
				07	90	0056E	MOVB	#7, IOB\$+44	
				5B	DD	00572	PUSHL	R11	
				7E	D4	00574	CLRL	-(SP)	
				5B	DD	00576	PUSHL	R8	
07	00000000G	EF	6A	03	FB	00578	CALLS	#3, XPO\$PUT	
	00000000G	EF		01	E1	0057F	BBC	#1, CMDBLK, 30\$	
				00	FB	00583	CALLS	#0, SPLIT	
			00000000G	EF	9F	0058A	PUSHAB	STR\$FAILURE	2049
				7E	D4	00590	CLRL	-(SP)	
			04FC	59	DD	00592	PUSHL	R9	
				C7	9F	00594	PUSHAB	\$STR\$STRING	
				7E	D4	00598	CLRL	-(SP)	
	00000000G	EF	50	05	FB	0059A	CALLS	#5, XST\$COPY	
				69	3C	005A1	MOVZWL	TMPSTR, R0	
	00000000G	EF	50	50	C0	005A4	ADDL2	R0, CHROUT	
	44	A8		69	9E	005AB	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	2C	A8		07	90	005AF	MOVB	#7, IOB\$+44	
				5B	DD	005B3	PUSHL	R11	
				7E	D4	005B5	CLRL	-(SP)	
				5B	DD	005B7	PUSHL	R8	
07	00000000G	EF	6A	03	FB	005B9	CALLS	#3, XPO\$PUT	
	00000000G	EF		01	E1	005C0	BBC	#1, CMDBLK, 31\$	
				00	FB	005C4	CALLS	#0, SPLIT	
			00000000G	EF	9F	005CB	PUSHAB	STR\$FAILURE	2050
				7E	D4	005D1	CLRL	-(SP)	
				59	DD	005D3	PUSHL	R9	
			0504	C7	9F	005D5	PUSHAB	\$STR\$STRING	
				7E	D4	005D9	CLRL	-(SP)	
	00000000G	EF	50	05	FB	005DB	CALLS	#5, XST\$COPY	
				69	3C	005E2	MOVZWL	TMPSTR, R0	
	00000000G	EF	50	50	C0	005E5	ADDL2	R0, CHROUT	
	44	A8		69	9E	005EC	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	2C	A8		07	90	005F0	MOVB	#7, IOB\$+44	
				5B	DD	005F4	PUSHL	R11	
				7E	D4	005F6	CLRL	-(SP)	
				5B	DD	005F8	PUSHL	R8	
07	00000000G	EF	6A	03	FB	005FA	CALLS	#3, XPO\$PUT	
	00000000G	EF		01	E1	00601	BBC	#1, CMDBLK, 32\$	
				00	FB	00605	CALLS	#0, SPLIT	
				7E	D4	0060C	CLRL	-(SP)	
			20	AA	DD	0060E	PUSHL	INDENTS	2060
			0903	8F	3C	00611	MOVZWL	#2307, -(SP)	
	00000000G	EF	53	03	FB	00616	CALLS	#3, XST\$ASCII	
				50	D0	0061D	MOVL	R0, R3	
				7E	D4	00620	CLRL	-(SP)	
				AA	DD	00622	PUSHL	CMDBLK+24	
			18	8F	3C	00625	MOVZWL	#2307, -(SP)	
	00000000G	EF	52	03	FB	0062A	CALLS	#3, XST\$ASCII	
				50	D0	00631	MOVL	R0, R2	
				7E	D4	00634	CLRL	-(SP)	
				AA	DD	00636	PUSHL	CMDBLK+24	
			18	8F	3C	00639	MOVZWL	#2307, -(SP)	
			0903						

		00000000G	EF	03	FB	0063E	CALLS	#3, XST\$ASCII	
				50	DD	00645	PUSHL	R0	
				051C	C7	9F	00647	PUSHAB	\$STR\$STRING4
					52	DD	0064B	PUSHL	R2
				0514	C7	9F	0064D	PUSHAB	\$STR\$STRING2
					53	DD	00651	PUSHL	R3
				050C	C7	9F	00653	PUSHAB	\$STR\$STRING0
		00000000G	EF	06	FB	00657	CALLS	#6, XST\$JOIN	
				00000000G	EF	9F	0065E	PUSHAB	STR\$FAILURE
					7E	D4	00664	CLRL	-(SP)
				0201	8F	BB	00666	PUSHR	#*M<R0,R9>
					7E	D4	0066A	CLRL	-(SP)
		00000000G	EF	05	FB	0066C	CALLS	#5, XST\$COPY	
			50	69	3C	00673	MOVZWL	TMPSTR, R0	
		00000000G	EF	50	C0	00676	ADDL2	R0, CHROUT	
			44	69	9E	0067D	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
			2C	07	90	00681	MOVB	#7, IOB\$+44	
					5B	DD	00685	PUSHL	R11
					7E	D4	00687	CLRL	-(SP)
					5B	DD	00689	PUSHL	R8
		00000000G	EF	03	FB	0068B	CALLS	#3, XPO\$PUT	
		07	6A	01	E1	00692	BBC	#1, CMDBLK, 34\$	
		00000000G	EF	00	FB	00696	CALLS	#0, SPLIT	
00F4	8F	00	6E	00	2C	0069D	MOVCS	#0, (SP), #0, #244, IOB\$	2067
					EF	006A4			
		00000000G	EF	8F	D0	006A9	MOVL	#50397245, IOB\$	
			0301003D	8F	B0	006B4	MOVW	#526, IOB\$RESULTANT+2	
		00000000G	EF	020E	8F	006B4	MOVW	#526, IOB\$RESULTANT+2	
			0524	C7	9E	006BD	MOVAB	\$IOB\$FILE SPEC, IOB\$+4	2070
		00000000G	EF	00010002	8F	C8	006C6	BISL2	#65538, IOB\$+46
			EF	01	90	006D1	MOVB	#1, IOB\$+44	
					EF	9F	006D3	PUSHAB	OPEN ERROR
					7E	D4	006DE	CLRL	-(SP)
					EF	9F	006E0	PUSHAB	IOB\$
		00000000G	EF	03	FB	006E6	CALLS	#3, XPO\$OPEN	
00F4	8F	00	6E	00	2C	006ED	MOVCS	#0, (SP), #0, #244, IOB\$	2074
					C7	006F4			
		00F8	C7	0301003D	8F	D0	006F7	MOVL	#50397245, IOB\$
		0116	C7	020E	8F	B0	00700	MOVW	#526, IOB\$RESULTANT+2
		00FC	C7	052C	C7	9E	00707	MOVAB	\$IOB\$FILE SPEC, IOB\$+4
		0126	C7	00010002	8F	C8	0070E	BISL2	#65538, IOB\$+46
		0124	C7		01	90	00717	MOVB	#1, IOB\$+44
					EF	9F	0071C	PUSHAB	OPEN ERROR
					7E	D4	00722	CLRL	-(SP)
					C7	9F	00724	PUSHAB	IOB\$
		00000000G	EF	03	FB	00728	CALLS	#3, XPO\$OPEN	
00F4	8F	00	6E	00	2C	0072F	MOVCS	#0, (SP), #0, #244, IOB\$	2081
					C7	00736			
		01EC	C7	0301003D	8F	D0	00739	MOVL	#50397245, IOB\$
		020A	C7	020E	8F	B0	00742	MOVW	#526, IOB\$RESULTANT+2
		01F0	C7	0534	C7	9E	00749	MOVAB	\$IOB\$FILE SPEC, IOB\$+4
		021A	C7	00010002	8F	C8	00750	BISL2	#65538, IOB\$+46
		0218	C7		01	90	00759	MOVB	#1, IOB\$+44
					EF	9F	0075E	PUSHAB	OPEN ERROR
					7E	D4	00764	CLRL	-(SP)
					C7	9F	00766	PUSHAB	IOB\$
		00000000G	EF	03	FB	0076A	CALLS	#3, XPO\$OPEN	
00F4	8F	00	6E	00	2C	00771	MOVCS	#0, (SP), #0, #244, IOB\$	2088

CO
VO

VO

																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							</
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

00000000G	EF	7E	D4	0086E	CLRL	-(SP)	
	50	05	FB	00870	CALLS	#5, XST\$COPY	
00000000G	EF	69	3C	00877	MOVZWL	TMPSTR, R0	
44	A8	50	C0	0087A	ADDL2	R0, CHROUT	
2C	A8	69	9E	00881	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		07	90	00885	MOVB	#7, IOB\$+44	
		5B	DD	00889	PUSHL	R11	
		7E	D4	0088B	CLRL	-(SP)	
		5B	DD	0088D	PUSHL	R8	
07 00000000G	EF	03	FB	0088F	CALLS	#3, XPOS\$PUT	
	6A	01	E1	00896	BBC	#1, CMDBLK, 40\$	
00000000G	EF	00	FB	0089A	CALLS	#0, SPLIT	
		EF	9F	008A1	PUSHAB	STR\$FAILURE	2122
		7E	D4	008A7	CLRL	-(SP)	
		59	DD	008A9	PUSHL	R9	
		C7	9F	008AB	PUSHAB	\$STR\$STRING	
		7E	D4	008AF	CLRL	-(SP)	
00000000G	EF	05	FB	008B1	CALLS	#5, XST\$COPY	
	50	69	3C	008B8	MOVZWL	TMPSTR, R0	
00000000G	EF	50	C0	008BB	ADDL2	R0, CHROUT	
44	A8	69	9E	008C2	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
2C	A8	07	90	008C6	MOVB	#7, IOB\$+44	
		5B	DD	008CA	PUSHL	R11	
		7E	D4	008CC	CLRL	-(SP)	
		5B	DD	008CE	PUSHL	R8	
07 00000000G	EF	03	FB	008D0	CALLS	#3, XPOS\$PUT	
	6A	01	E1	008D7	BBC	#1, CMDBLK, 41\$	
00000000G	EF	00	FB	008DB	CALLS	#0, SPLIT	
34	A7	08	B0	008E2	MOVW	#8, IOB\$+52	2132
36	A7	02	90	008E6	MOVB	#2, IOB\$+54	
2C	A7	06	90	008EA	MOVB	#6, IOB\$+44	
		5B	DD	008EE	PUSHL	R11	
		7E	D4	008F0	CLRL	-(SP)	
		57	DD	008F2	PUSHL	R7	
00000000G	EF	03	FB	008F4	CALLS	#3, XPOS\$GET	
00208001	8F	50	D1	008FB	CMPL	R0, #2129921	
		0D	13	00902	BEQL	42\$	
		A7	9F	00904	PUSHAB	TOCIOB+28	2140
		01	DD	00907	PUSHL	#1	
		8F	DD	00909	PUSHL	#DSRTOC\$_EMPTYIN	
		3B	11	0090F	BRB	47\$	
	50	A7	D0	00911	MOVL	TOCIOB+56, R0	2155
	51	01	D0	00915	MOVL	#1, CORRECT_FORMAT	2160
FFFFFFFF	8F	60	D1	00918	CMPL	(R0), #-1	2165
		0B	12	0091F	BNEQ	43\$	
00F4	C7	01	CE	00921	MNEGL	#1, FILE_TYPE	2170
	02	A0	D1	00926	CMPL	4(R0), #2	2172
		0E	11	0092A	BRB	44\$	
	01	60	D1	0092C	CMPL	(R0), #1	2175
		0B	12	0092F	BNEQ	45\$	
00F4	C7	01	D0	00931	MOVL	#1, FILE_TYPE	2180
	03	A0	D1	00936	CMPL	4(R0), #3	2182
		02	13	0093A	BEQL	46\$	
		51	D4	0093C	CLRL	CORRECT_FORMAT	2189
		51	E8	0093E	BLBS	CORRECT_FORMAT, 48\$	2193
	14	A7	9F	00941	PUSHAB	TOCIOB+28	2201
		01	DD	00944	PUSHL	#1	

CONTENTS
V04-000

CONTENTS - Table of Contents formatter program
TOCINI - Initialization

G 12

16-Sep-1984 00:01:05

14-Sep-1984 13:05:48

VAX-11 Bliss-32 V4.0-742

[RUNOFF.SRC]CONTENTS.BLI;1

Page 64

(3)

CO

VO

00000000G	00	00000000G	8F	DD	00946		PUSHL	#DSRTOCS	INVINPUT
			03	FB	0094C	47\$:	CALLS	#3	LIB\$SIGNAL
			04	11	00953		BRB	49\$	
	50		01	D0	00955	48\$:	MOVL	#1,	R0
				04	00958		RET		
			50	D4	00959	49\$:	CLRL	R0	
				04	0095B		RET		

: 2208
: 2213
: 2214
:

; Routine Size: 2396 bytes, Routine Base: \$CODE\$ + 10FE

```

1584 2215 1 %SBTTL 'PROCESS_PAGE - Process page reference record'
1585 2216 1 ROUTINE process_page : NOVALUE =
1586 2217 1 ++
1587 2218 1
1588 2219 1 FUNCTIONAL DESCRIPTION:
1589 2220 1
1590 2221 1 This routine is called by TOC () to process a page reference record.
1591 2222 1
1592 2223 1 FORMAL PARAMETERS:
1593 2224 1
1594 2225 1 None
1595 2226 1
1596 2227 1 IMPLICIT INPUTS:
1597 2228 1
1598 2229 1 None
1599 2230 1
1600 2231 1 IMPLICIT OUTPUTS:
1601 2232 1
1602 2233 1 pagen - contains the new page reference
1603 2234 1 The page reference record is written to the four temporary files.
1604 2235 1
1605 2236 1 ROUTINE VALUE:
1606 2237 1 COMPLETION CODES:
1607 2238 1
1608 2239 1 None
1609 2240 1
1610 2241 1 SIDE EFFECTS:
1611 2242 1
1612 2243 1 None
1613 2244 1 --
1614 2245 2 BEGIN
1615 2246 2 errcnt = 0; ! Reset count of CONFUSED messages.
1616 2247 2
1617 2248 2 |
1618 2249 2 | Read a page number.
1619 2250 2 |
1620 2251 2 $xpo_get (iob = tociob, fullwords = page_sct_size);
1621 2252 2
1622 2253 3 BEGIN
1623 2254 3
1624 2255 3 BIND
1625 2256 3 page_ref = .toclob [iob$a_data] : VECTOR [page_sct_size];
1626 2257 3
1627 2258 3 MAP
1628 2259 3 pagen : VECTOR [page_sct_size];
1629 2260 3
1630 2261 3 INCR i FROM 1 TO page_sct_size DO
1631 2262 4 BEGIN
1632 2263 4 |
1633 2264 4 | Copy page number
1634 2265 4 |
1635 2266 4 pagen [.i - 1] = .page_ref [.i - 1];
1636 2267 3 END;
1637 2268 3
1638 2269 2 END;
1639 2270 2
1640 2271 3 BEGIN

```

```

: 1641 2272 3
: 1642 2273 3
: 1643 2274 3
: 1644 2275 3
: 1645 2276 3
: 1646 2277 3
: 1647 2278 3
: 1648 2279 3
: 1649 2280 3
: 1650 2281 3
: 1651 2282 3
: 1652 2283 3
: 1653 2284 3
: 1654 2285 3
: 1655 2286 3
: 1656 2287 3
: 1657 2288 3
: 1658 2289 3
: 1659 2290 1

```

```

Write page information to temp files

LOCAL
    record_header : VECTOR [2];

record_header [0] = maj_runoff;
record_header [1] = min_page;
$spo_put (iob = stciob, binary_data = (2, record_header));
$spo_put (iob = stciob, binary_data = (page_sct_size, pagen));
$spo_put (iob = fgciob, binary_data = (2, record_header));
$spo_put (iob = fgciob, binary_data = (page_sct_size, pagen));
$spo_put (iob = tbiob, binary_data = (2, record_header));
$spo_put (iob = tbiob, binary_data = (page_sct_size, pagen));
$spo_put (iob = tbiob, binary_data = (page_sct_size, pagen));
$spo_put (iob = exiob, binary_data = (2, record_header));
$spo_put (iob = exiob, binary_data = (page_sct_size, pagen));
END;
END;

```

```

IOB$=
IOB$=
IOB$=
IOB$=
IOB$=
IOB$=
IOB$=
IOB$=
TOCJOB
FGJOB
FGJOB
TBJOB
TBJOB
EXJOB
EXJOB

```

```

007C 00000 PROCESS_PAGE:
56 00000000G EF 9E 00002 .WORD Save R2,R3,R4,R5,R6
55 00000000G EF 9E 00009 MOVAB PAGEN, R6
54 00000000G EF 9E 00010 MOVAB IOB$+68, R5
53 00000000G EF 9E 00017 MOVAB XPOS$PUT, R4
52 00000000' EF 9E 0001E MOVAB XPOS$FAILURE, R3
5E 10 C2 00025 MOVAB IOB$+52, R2
03B4 C2 D4 00028 SUBL2 #16, SP
62 10 B0 0002C CLRL ERRCNT
02 A2 02 90 0002F MOVW #16, IOB$+52
F8 A2 06 90 00033 MOVW #2, IOB$+54
53 DD 00037 MOVW #6, IOB$+44
7E D4 00039 PUSHL R3
CC A2 9F 0003B CLRL -(SP)
03 FB 0003E PUSHAB IOB$
51 01 D0 00045 CALLS #3, XPOS$GET
50 04 B241 DE 00048 1$: MOVW #1, I
FC A641 FC A0 D0 0004D MOVW @TOCJOB+56[I], R0
F1 51 04 F3 00053 MOVW -4(R0), PAGEN-4[I]
08 AE 02 D0 00057 AOBLEQ #4, I, 1$
0C AE 01 D0 0005B MOVW #2, RECORD_HEADER
6E 08 B0 0005F MOVW #1, RECORD_HEADER+4
02 AE 02 90 00062 MOVW #8, $IOB$OUTPUT
03 AE 01 90 00066 MOVW #2, $IOB$OUTPUT+2
04 AE 08 AE 9E 0006A MOVW #1, $IOB$OUTPUT+3
MOVAB RECORD_HEADER, $IOB$OUTPUT+4

```

E8	65 A5		6E 9E 0006F	MOVAB \$IOB\$OUTPUT, IOB\$+68	
			07 90 00072	MOVAB #7, IOB\$+44	
			53 DD 00076	PUSHL R3	
			7E D4 00078	CLRL -(SP)	
		BC	A5 9F 0007A	PUSHAB IOB\$	
	64		03 FB 0007D	CALLS #3, XPOS\$PUT	
	6E		10 B0 00080	MOVW #16, \$IOB\$OUTPUT	2281
02	AE		02 90 00083	MOVAB #2, \$IOB\$OUTPUT+2	
03	AE		01 90 00087	MOVAB #1, \$IOB\$OUTPUT+3	
04	AE		66 9E 0008B	MOVAB PAGEN, \$IOB\$OUTPUT+4	
	65		6E 9E 0008F	MOVAB \$IOB\$OUTPUT, IOB\$+68	
E8	A5		07 90 00092	MOVAB #7, IOB\$+44	
			53 DD 00096	PUSHL R3	
			7E D4 00098	CLRL -(SP)	
		BC	A5 9F 0009A	PUSHAB IOB\$	
	64		03 FB 0009D	CALLS #3, XPOS\$PUT	
	6E		08 B0 000A0	MOVW #8, \$IOB\$OUTPUT	2282
02	AE		02 90 000A3	MOVAB #2, \$IOB\$OUTPUT+2	
03	AE		01 90 000A7	MOVAB #1, \$IOB\$OUTPUT+3	
04	AE	08	AE 9E 000AB	MOVAB RECORD HEADER, \$IOB\$OUTPUT+4	
0108	C2		6E 9E 000B0	MOVAB \$IOB\$OUTPUT, IOB\$+68	
00F0	C2		07 90 000B5	MOVAB #7, IOB\$+44	
			53 DD 000BA	PUSHL R3	
			7E D4 000BC	CLRL -(SP)	
		00C4	C2 9F 000BE	PUSHAB IOB\$	
	64		03 FB 000C2	CALLS #3, XPOS\$PUT	
	6E		10 B0 000C5	MOVW #16, \$IOB\$OUTPUT	2283
02	AE		02 90 000C8	MOVAB #2, \$IOB\$OUTPUT+2	
03	AE		01 90 000CC	MOVAB #1, \$IOB\$OUTPUT+3	
04	AE		66 9E 000D0	MOVAB PAGEN, \$IOB\$OUTPUT+4	
0108	C2		6E 9E 000D4	MOVAB \$IOB\$OUTPUT, IOB\$+68	
00F0	C2		07 90 000D9	MOVAB #7, IOB\$+44	
			53 DD 000DE	PUSHL R3	
			7E D4 000E0	CLRL -(SP)	
		00C4	C2 9F 000E2	PUSHAB IOB\$	
	64		03 FB 000E6	CALLS #3, XPOS\$PUT	
	6E		08 B0 000E9	MOVW #8, \$IOB\$OUTPUT	2284
02	AE		02 90 000EC	MOVAB #2, \$IOB\$OUTPUT+2	
03	AE		01 90 000F0	MOVAB #1, \$IOB\$OUTPUT+3	
04	AE	08	AE 9E 000F4	MOVAB RECORD HEADER, \$IOB\$OUTPUT+4	
01FC	C2		6E 9E 000F8	MOVAB \$IOB\$OUTPUT, IOB\$+68	
01E4	C2		07 90 000FE	MOVAB #7, IOB\$+44	
			53 DD 00103	PUSHL R3	
			7E D4 00105	CLRL -(SP)	
		01B8	C2 9F 00107	PUSHAB IOB\$	
	64		03 FB 0010B	CALLS #3, XPOS\$PUT	
	6E		10 B0 0010E	MOVW #16, \$IOB\$OUTPUT	2285
02	AE		02 90 00111	MOVAB #2, \$IOB\$OUTPUT+2	
03	AE		01 90 00115	MOVAB #1, \$IOB\$OUTPUT+3	
04	AE		66 9E 00119	MOVAB PAGEN, \$IOB\$OUTPUT+4	
01FC	C2		6E 9E 0011D	MOVAB \$IOB\$OUTPUT, IOB\$+68	
01E4	C2		07 90 00122	MOVAB #7, IOB\$+44	
			53 DD 00127	PUSHL R3	
			7E D4 00129	CLRL -(SP)	
		01B8	C2 9F 0012B	PUSHAB IOB\$	
	64		03 FB 0012F	CALLS #3, XPOS\$PUT	
	6E		08 B0 00132	MOVW #8, \$IOB\$OUTPUT	2286

02	AE		02	90	00135
03	AE		01	90	00139
04	AE	08	AE	9E	0013D
02F0	C2		6E	9E	00142
02D8	C2		07	90	00147
			53	DD	0014C
			7E	D4	0014E
		02AC	C2	9F	00150
	64		03	FB	00154
	6E		10	B0	00157
02	AE		02	90	0015A
03	AE		01	90	0015E
04	AE		66	9E	00162
02F0	C2		6E	9E	00166
02D8	C2		07	90	0016B
			53	DD	00170
			7E	D4	00172
		02AC	C2	9F	00174
	64		03	FB	00178
			04	0017B	

```
MOVB #2, $IOB$OUTPUT+2
MOVB #1, $IOB$OUTPUT+3
MOVAB RECORD HEADER, $IOB$OUTPUT+4
MOVAB $IOB$OUTPUT, IOB$+68
MOVB #7, IOB$+44
PUSHL R3
CLRL -(SP)
PUSHAB IOB$
CALLS #3, XPO$PUT
MOVW #16, $IOB$OUTPUT
MOVB #2, $IOB$OUTPUT+2
MOVB #1, $IOB$OUTPUT+3
MOVAB PAGEN, $IOB$OUTPUT+4
MOVAB $IOB$OUTPUT, IOB$+68
MOVB #7, IOB$+44
PUSHL R3
CLRL -(SP)
PUSHAB IOB$
CALLS #3, XPO$PUT
RET
```

2287

2290

; Routine Size: 380 bytes, Routine Base: \$CODE\$ + 1A5A

```

1661 2291 1 %SBTTL 'PROCESS_ENTITY_INFO - Process figure/table/example info record'
1662 2292 1 ROUTINE process_entity_info : NOVALUE =
1663 2293 1 ++
1664 2294 1
1665 2295 1 FUNCTIONAL DESCRIPTION:
1666 2296 1
1667 2297 1 This routine is called by TOC () to process a figure/table/example
1668 2298 1 information record.
1669 2299 1
1670 2300 1 FORMAL PARAMETERS:
1671 2301 1
1672 2302 1 None
1673 2303 1
1674 2304 1 IMPLICIT INPUTS:
1675 2305 1
1676 2306 1 major - major record code
1677 2307 1 minor - minor record code
1678 2308 1 exiob - IOB for examples temp file
1679 2309 1 fgiob - IOB for figures temp file
1680 2310 1 tbiob - IOB for tables temp file
1681 2311 1
1682 2312 1 IMPLICIT OUTPUTS:
1683 2313 1
1684 2314 1 The record is written to the appropriate temp file if new captions
1685 2315 1 are being processed.
1686 2316 1
1687 2317 1 ROUTINE VALUE:
1688 2318 1 COMPLETION CODES:
1689 2319 1
1690 2320 1 None
1691 2321 1
1692 2322 1 SIDE EFFECTS:
1693 2323 1
1694 2324 1 None
1695 2325 1 --
1696 2326 2 BEGIN
1697 2327 2
1698 2328 2 LOCAL
1699 2329 2 prefix_addr,
1700 2330 2 prefix_fullwords,
1701 2331 2 suffix_addr,
1702 2332 2 suffix_fullwords,
1703 2333 2 iob_ptr : REF $xpo_iob (),
1704 2334 2 info_list : VECTOR[6];
1705 2335 2
1706 2336 2 errcnt = 0; ! Reset count of CONFUSED messages.
1707 2337 3 iob_ptr = (SELECTONE .minor OF
1708 2338 3 SET
1709 2339 3 [min_examp_inf] : exiob;
1710 2340 3 [min_table_inf] : tbiob;
1711 2341 3 [min_figur_inf] : fgiob;
1712 2342 3 TES);
1713 2343 2
1714 2344 2 info_list [0] = .major;
1715 2345 2 info_list [1] = .minor;
1716 2346 2
1717 2347 2 ! Get counter, display descriptor and prefix text length

```

```
1718 2348 2 !
1719 2349 2 $xpo_get (iob = tociob, fullwords = 3);
1720 2350 2 BEGIN
1721 2351 2
1722 2352 2 BIND
1723 2353 2 counter_info = .toclob [iob$a_data] : VECTOR;
1724 2354 2
1725 2355 2 info_list [2] = .counter_info [0]; ! Counter value
1726 2356 2 info_list [3] = .counter_info [1]; ! Counter display code
1727 2357 2 info_list [4] = .counter_info [2]; ! Prefix text length
1728 2358 2 END;
1729 2359 2
1730 2360 2 IF .info_list [4] NEQ 0
1731 2361 2 THEN
1732 2362 2 BEGIN
1733 2363 2     There is prefix text
1734 2364 2
1735 2365 2     prefix_fullwords = CH$ALLOCATION (.info_list [4]);
1736 2366 2     $xpo_get (iob = tociob, fullwords = .prefix_fullwords);
1737 2367 2
1738 2368 2     Get some memory to store the prefix string in.
1739 2369 2
1740 2370 2     $xpo_get_mem (fullwords = .prefix_fullwords, result = prefix_addr);
1741 2371 2
1742 2372 2     Copy the prefix string to the memory just allocated
1743 2373 2
1744 2374 2 BEGIN
1745 2375 2 BIND
1746 2376 2     input_vector = .toclob [iob$a_data] : VECTOR,
1747 2377 2     prefix_vector = .prefix_addr : VECTOR;
1748 2378 2
1749 2379 2     INCR i FROM 0 TO .prefix_fullwords - 1 DO
1750 2380 2         prefix_vector [i] = .input_vector [i];
1751 2381 2
1752 2382 2     END;
1753 2383 2 END;
1754 2384 2
1755 2385 2
1756 2386 2 !
1757 2387 2 ! Get length of suffix text
1758 2388 2
1759 2389 2 $xpo_get (iob = tociob, fullwords = 1);
1760 2390 2 info_list [5] = (.toclob [iob$a_data]);
1761 2391 2
1762 2392 2 IF .info_list [5] NEQ 0
1763 2393 2 THEN
1764 2394 2 BEGIN
1765 2395 2     There is suffix text
1766 2396 2
1767 2397 2     suffix_fullwords = CH$ALLOCATION (.info_list [5]);
1768 2398 2     $xpo_get (iob = tociob, fullwords = .suffix_fullwords);
1769 2399 2
1770 2400 2     Get some memory to store the suffix string in.
1771 2401 2
1772 2402 2     $xpo_get_mem (fullwords = .suffix_fullwords, result = suffix_addr);
1773 2403 2
1774 2404 2
```



```

: 1775      2405      3      ! Copy the suffix string to the dynamic memory area
: 1776      2406      3
: 1777      2407      4      BEGIN
: 1778      2408      4      BIND
: 1779      2409      4      input_vector = .tociob [iob$a_data] : VECTOR,
: 1780      2410      4      suffix_vector = .suffix_addr : VECTOR;
: 1781      2411      4
: 1782      2412      4      INCR i FROM 0 TO .suffix_fullwords - 1 DO
: 1783      2413      4      suffix_vector [.i] = .input_vector [.i];
: 1784      2414      4
: 1785      2415      3      END;
: 1786      2416      2      END;
: 1787      2417      2
: 1788      2418      2      IF vfycap (new_captions)
: 1789      2419      2      THEN
: 1790      2420      2      BEGIN
: 1791      2421      2
: 1792      2422      2      Write major-code, minor-code, counter,
: 1793      2423      2      display-descriptor and prefix text length.
: 1794      2424      2
: 1795      2425      2      $xpo_put (iob = .iob_ptr, binary_data = (5, info_list));
: 1796      2426      2
: 1797      2427      2      Write prefix text if any
: 1798      2428      2
: 1799      2429      2      IF .info_list [4] NEQ 0
: 1800      2430      2      THEN
: 1801      2431      2      $xpo_put (iob = .iob_ptr,
: 1802      2432      2      binary_data = (.prefix_fullwords, .prefix_addr));
: 1803      2433      2
: 1804      2434      2
: 1805      2435      2      Write suffix text length
: 1806      2436      2
: 1807      2437      2      $xpo_put (iob = .iob_ptr, binary_data = (1, info_list [5]));
: 1808      2438      2
: 1809      2439      2      Write suffix text if any
: 1810      2440      2
: 1811      2441      2      IF .info_list [5] NEQ 0
: 1812      2442      2      THEN
: 1813      2443      2      $xpo_put (iob = .iob_ptr,
: 1814      2444      2      binary_data = (.suffix_fullwords, .suffix_addr));
: 1815      2445      2
: 1816      2446      2      END;
: 1817      2447      2
: 1818      2448      2      IF .info_list [4] NEQ 0
: 1819      2449      2      THEN
: 1820      2450      2      BEGIN
: 1821      2451      2
: 1822      2452      2      Deallocate the memory we allocated for the prefix string
: 1823      2453      2
: 1824      2454      2      $xpo_free_mem (binary_data = (.prefix_fullwords, .prefix_addr));
: 1825      2455      2      END;
: 1826      2456      2
: 1827      2457      2      IF .info_list [5] NEQ 0
: 1828      2458      2      THEN
: 1829      2459      2      BEGIN
: 1830      2460      2
: 1831      2461      2      Deallocate the memory allocated for the suffix string

```

```

: 1832
: 1833
: 1834
: 1835
: 1836
2462 3
2463 3
2464 2
2465 2
2466 1
END:

```

```

!
$ xpo_free_mem (binary_data = (.suffix_fullwords, .suffix_addr));
END;

```

```

IOB$= TOC IOB
IOB$= TOC IOB
IOB$= TOC IOB
IOB$= TOC IOB
      .EXTRN XPOS$ALLOC_MEM, XPOS$FREE_MEM

```

OFFC 00000 PROCESS_ENTITY_INFO:

	5B	00000000G	EF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 2292
	5A	00000000G	EF	9E	00009	MOVAB	XPOS\$PUT, R11	
	59	00000000G	EF	9E	00010	MOVAB	XPOS\$GET, R10	
	58	00000000'	EF	9E	00017	MOVAB	XPOS\$FAILURE, R9	
	5E		20	C2	0001E	MOVAB	IOB\$+52, R8	
		03B4	C8	D4	00021	SUBL2	#32, SP	
	50	00000000G	EF	D0	00025	CLRL	ERRCNT	2336
	07		50	D1	0002C	MOVL	MINOR, R0	2337
			07	12	0002F	CMPL	R0, #7	2339
	54	02AC	C8	9E	00031	BNEQ	1\$	
			1B	11	00036	MOVAB	EXIOB, IOB_PTR	
	0D		50	D1	00038	BRB	4\$	2340
			07	12	0003B	CMPL	R0, #13	
	54	01B8	C8	9E	0003D	BNEQ	2\$	
			0F	11	00042	MOVAB	TBIOB, IOB_PTR	
	0A		50	D1	00044	BRB	4\$	2341
			05	13	00047	CMPL	R0, #10	
	54		01	CE	00049	BEQL	3\$	
			05	11	0004C	MNEGL	#1, IOB_PTR	
	54	00C4	C8	9E	0004E	BRB	4\$	
08	AE	00000000G	EF	D0	00053	MOVAB	FGIOB, IOB_PTR	2344
0C	AE		50	D0	0005B	MOVL	MAJOR, INFO_LIST	2345
	68		0C	B0	0005F	MOVL	R0, INFO_LIST+4	2349
02	A8		02	90	00062	MOVW	#12, IOB\$+52	
F8	A8		06	90	00066	MOVB	#2, IOB\$+54	
			59	DD	0006A	MOVB	#6, IOB\$+44	
			7E	D4	0006C	PUSHL	R9	
		CC	A8	9F	0006E	CLRL	-(SP)	
	6A		03	FB	00071	PUSHAB	IOB\$	
	50	04	A8	D0	00074	CALLS	#3, XPOS\$GET	2353
10	AE		60	7D	00078	MOVL	TOC IOB+56, R0	2355
18	AE	08	A0	D0	0007C	MOVQ	(R0), INFO_LIST+8	2357
			55	13	00081	MOVL	8(R0), INFO_LIST+16	2360
50	18	AE	03	C1	00083	BEQL	8\$	2366
53		50	04	C7	00088	ADDL3	#3, INFO_LIST+16, R0	
52		53	02	78	0008C	DIVL3	#4, R0, PREFIX_FULLWORDS	2367
		68	52	B0	00090	ASHL	#2, PREFIX_FULLWORDS, R2	
02	A8		02	90	00093	MOVW	R2, IOB\$+52	
F8	A8		06	90	00097	MOVB	#2, IOB\$+54	
			59	DD	0009B	MOVB	#6, IOB\$+44	
			7E	D4	0009D	PUSHL	R9	
						CLRL	-(SP)	

		CC	A8	9F	0009F	PUSHAB	IOB\$		
	6A		03	FB	000A2	CALLS	#3, XPOSGET		
	6E	02020000	8F	DO	000A5	MOVL	#33685504, \$XPOSDESC	2371	
		04	AE	D4	000AC	CLRL	\$XPOSDESC+4		
			59	DD	000AF	PUSHL	R9		
	7E		7E	7C	000B1	CLRQ	-(SP)		
		10	01	CE	000B3	MNEGL	#1, -(SP)		
			AE	9F	000B6	PUSHAB	\$XPOSDESC		
			52	DD	000B9	PUSHL	R2		
00000000G	EF		06	FB	000BB	CALLS	#6, XPOSALLOC MEM		
	04		50	E9	000C2	BLBC	\$XPOSSTATUS, 5\$		
	56	04	AE	DO	000C5	MOVL	\$XPOSDESC+4, PREFIX_ADDR	2380	
	50		01	CE	000C9	MNEGL	#1, I		
			06	11	000CC	BRB	7\$		
	6640	04	B8	40	DO	000CE	MOVL	@TOCIOB+56[I], (PREFIX_ADDR)[I]	2381
F6	50		53	F2	000D4	AOBLSS	PREFIX_FULLWORDS, I, 6\$		
	68		04	B0	000D8	MOVW	#4, IOB\$+52	2389	
	02	A8	02	90	000DB	MOVB	#2, IOB\$+54		
F8	A8		06	90	000DF	MOVB	#6, IOB\$+44		
			59	DD	000E3	PUSHL	R9		
			7E	D4	000E5	CLRL	-(SP)		
		CC	A8	9F	000E7	PUSHAB	IOB\$		
	6A		03	FB	000EA	CALLS	#3, XPOSGET		
1C	AE	04	B8	DO	000ED	MOVL	@TOCIOB+56, INFO_LIST+20	2390	
			57	D4	000F2	CLRL	R7	2392	
		1C	AE	D5	000F4	TSTL	INFO_LIST+20		
			57	13	000F7	BEQL	12\$		
			57	D6	000F9	INCL	R7		
50	1C	AE	03	C1	000FB	ADDL3	#3, INFO_LIST+20, R0	2398	
52		50	04	C7	00100	DIVL3	#4, R0, SUFFIX_FULLWORDS		
55		52	02	78	00104	ASHL	#2, SUFFIX_FULLWORDS, R5	2399	
		68	55	B0	00108	MOVW	R5, IOB\$+52		
	02	A8	02	90	0010B	MOVB	#2, IOB\$+54		
F8	A8		06	90	0010F	MOVB	#6, IOB\$+44		
			59	DD	00113	PUSHL	R9		
			7E	D4	00115	CLRL	-(SP)		
		CC	A8	9F	00117	PUSHAB	IOB\$		
	6A		03	FB	0011A	CALLS	#3, XPOSGET		
	6E	02020000	8F	DO	0011D	MOVL	#33685504, \$XPOSDESC	2403	
		04	AE	D4	00124	CLRL	\$XPOSDESC+4		
			59	DD	00127	PUSHL	R9		
			7E	7C	00129	CLRQ	-(SP)		
	7E		01	CE	0012B	MNEGL	#1, -(SP)		
		10	AE	9F	0012E	PUSHAB	\$XPOSDESC		
			55	DD	00131	PUSHL	R5		
00000000G	EF		06	FB	00133	CALLS	#6, XPOSALLOC MEM		
	04		50	E9	0013A	BLBC	\$XPOSSTATUS, 9\$		
	55	04	AE	DO	0013D	MOVL	\$XPOSDESC+4, SUFFIX_ADDR	2412	
	50		01	CE	00141	MNEGL	#1, I		
			06	11	00144	BRB	11\$		
	6540	04	B8	40	DO	00146	MOVL	@TOCIOB+56[I], (SUFFIX_ADDR)[I]	2413
F6	50		52	F2	0014C	AOBLSS	SUFFIX_FULLWORDS, I, 10\$		
			01	DD	00150	PUSHL	#1	2418	
00000000V	EF		01	FB	00152	CALLS	#1, VFYCAP		
	03		50	E8	00159	BLBS	R0, 13\$		
		008C	31	0015C	BRW	15\$			
	6E		14	B0	0015F	MOVW	#20, \$IOB\$OUTPUT	2425	

02	AE	02	90	00162	MOVB	#2, \$IOB\$OUTPUT+2	
03	AE	01	90	00166	MOVB	#1, \$IOB\$OUTPUT+3	
04	AE	08	AE	9E 0016A	MOVAB	INFO_LIST, \$IOB\$OUTPUT+4	
44	A4		6E	9E 0016F	MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
2C	A4		07	90 00173	MOVB	#7, 44(IOB_PTR)	
			59	DD 00177	PUSHL	R9	
			7E	D4 00179	CLRL	-(SP)	
			54	DD 0017B	PUSHL	IOB_PTR	
	6B		03	FB 0017D	CALLS	#3, XPOS\$PUT	
		18	AE	D5 00180	TSTL	INFO_LIST+16	2429
			21	13 00183	BEQL	14\$	
6E	53		04	A5 00185	MULW3	#4, PREFIX_FULLWORDS, \$IOB\$OUTPUT	2432
02	AE		02	90 00189	MOVB	#2, \$IOB\$OUTPUT+2	
03	AE		01	90 0018D	MOVB	#1, \$IOB\$OUTPUT+3	
04	AE		56	D0 00191	MOVL	PREFIX_ADDR, \$IOB\$OUTPUT+4	
44	A4		6E	9E 00195	MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
2C	A4		07	90 00199	MOVB	#7, 44(IOB_PTR)	
			59	DD 0019D	PUSHL	R9	
			7E	D4 0019F	CLRL	-(SP)	
			54	DD 001A1	PUSHL	IOB_PTR	
	6B		03	FB 001A3	CALLS	#3, XPOS\$PUT	
02	AE		04	B0 001A6	MOVW	#4, \$IOB\$OUTPUT	2437
03	AE		02	90 001A9	MOVB	#2, \$IOB\$OUTPUT+2	
04	AE	1C	01	90 001AD	MOVB	#1, \$IOB\$OUTPUT+3	
44	A4		AE	9E 001B1	MOVAB	INFO_LIST+20, \$IOB\$OUTPUT+4	
2C	A4		6E	9E 001B6	MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
			07	90 001BA	MOVB	#7, 44(IOB_PTR)	
			59	DD 001BE	PUSHL	R9	
			7E	D4 001C0	CLRL	-(SP)	
			54	DD 001C2	PUSHL	IOB_PTR	
	6B		03	FB 001C4	CALLS	#3, XPOS\$PUT	
	21		57	E9 001C7	BLBC	R7, 15\$	2441
6E	52		04	A5 001CA	MULW3	#4, SUFFIX_FULLWORDS, \$IOB\$OUTPUT	2444
02	AE		02	90 001CE	MOVB	#2, \$IOB\$OUTPUT+2	
03	AE		01	90 001D2	MOVB	#1, \$IOB\$OUTPUT+3	
04	AE		55	D0 001D6	MOVL	SUFFIX_ADDR, \$IOB\$OUTPUT+4	
44	A4		6E	9E 001DA	MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
2C	A4		07	90 001DE	MOVB	#7, 44(IOB_PTR)	
			59	DD 001E2	PUSHL	R9	
			7E	D4 001E4	CLRL	-(SP)	
			54	DD 001E6	PUSHL	IOB_PTR	
	6B		03	FB 001E8	CALLS	#3, XPOS\$PUT	
		18	AE	D5 001EB	TSTL	INFO_LIST+16	2448
			21	13 001EE	BEQL	16\$	
6E	53		04	A5 001F0	MULW3	#4, PREFIX_FULLWORDS, \$XPOS\$DESC	2454
02	AE		02	90 001F4	MOVB	#2, \$XPOS\$DESC+2	
03	AE		02	90 001F8	MOVB	#2, \$XPOS\$DESC+3	
04	AE		56	D0 001FC	MOVL	PREFIX_ADDR, \$XPOS\$DESC+4	
			59	DD 00200	PUSHL	R9	
			7E	7C 00202	CLRL	-(SP)	
	7E		02	CE 00204	MNEGL	#2, -(SP)	
00000000G	EF	10	AE	9F 00207	PUSHAB	\$XPOS\$DESC	
	21		05	FB 0020A	CALLS	#5, XPOS\$FREE_MEM	
6E	52		57	E9 00211	BLBC	R7, 17\$	2457
02	AE		04	A5 00214	MULW3	#4, SUFFIX_FULLWORDS, \$XPOS\$DESC	2463
03	AE		02	9C 00218	MOVB	#2, \$XPOS\$DESC+2	
			02	90 0021C	MOVB	#2, \$XPOS\$DESC+3	

```
; Routine Size: 566 bytes,    Routine Base: $CODE$ + 1BD6
```

```

1838 2467 1 %SBTTL 'PROCESS_ENTITY_TXT - Process figure/table/example text record'
1839 2468 1 ROUTINE process_entity_txt : NOVALUE =
1840 2469 1 ++
1841 2470 1
1842 2471 1 FUNCTIONAL DESCRIPTION:
1843 2472 1
1844 2473 1 This routine is called by TOC () to process a figure/table/example
1845 2474 1 text record.
1846 2475 1
1847 2476 1 FORMAL PARAMETERS:
1848 2477 1
1849 2478 1 None
1850 2479 1
1851 2480 1 IMPLICIT INPUTS:
1852 2481 1
1853 2482 1 major - major record code
1854 2483 1 minor - minor record code
1855 2484 1 exiob - IOB for examples temp file
1856 2485 1 fgiob - IOB for figures temp file
1857 2486 1 tbiob - IOB for tables temp file
1858 2487 1
1859 2488 1 IMPLICIT OUTPUTS:
1860 2489 1
1861 2490 1 txtlen - length of caption text
1862 2491 1
1863 2492 1 The record is written to the appropriate temp file if new captions
1864 2493 1 are being processed.
1865 2494 1
1866 2495 1 ROUTINE VALUE:
1867 2496 1 COMPLETION CODES:
1868 2497 1
1869 2498 1 None
1870 2499 1
1871 2500 1 SIDE EFFECTS:
1872 2501 1
1873 2502 1 None
1874 2503 1 --
1875 2504 2 BEGIN
1876 2505 2
1877 2506 2 LOCAL
1878 2507 2 iob_ptr : REF $xpo_iob ()
1879 2508 2 record_header : VECTOR [3];
1880 2509 2
1881 2510 2 errcnt = 0; ! Reset count of CONFUSED messages.
1882 2511 2
1883 2512 3 iob_ptr = (SELECTONE .minor OF
1884 2513 3 SET
1885 2514 3 [min_figur_txt] : fgiob;
1886 2515 3 [min_figur_txt_u] : fgiob;
1887 2516 3 [min_table_txt] : tbiob;
1888 2517 3 [min_table_txt_u] : tbiob;
1889 2518 3 [min_examp_txt] : exiob;
1890 2519 3 [min_examp_txt_u] : exiob;
1891 2520 2 TES);
1892 2521 2
1893 2522 2 !
1894 2523 2 ! Get length of text

```

```

: 1895 2524 2 !
: 1896 2525 2 $xpo_get (iob = tociob, fullwords = 1);
: 1897 2526 2 txtlen = .(toclob [iob$a_data]);
: 1898 2527 2
: 1899 2528 2 record_header [0] = .major;
: 1900 2529 2 record_header [1] = .minor;
: 1901 2530 2 record_header [2] = .txtlen;
: 1902 2531 2
: 1903 2532 2 ! Get the text, if there is any
: 1904 2533 2
: 1905 2534 2 IF .txtlen NEQ 0
: 1906 2535 2 THEN
: 1907 2536 2 $xpo_get (iob = tociob, fullwords = CH$ALLOCATION (.txtlen));
: 1908 2537 2
: 1909 2538 2 IF vfyap (new_captions)
: 1910 2539 2 THEN
: 1911 2540 2 BEGIN
: 1912 2541 2 !
: 1913 2542 2 ! Write record to temp file
: 1914 2543 2
: 1915 2544 2 $xpo_put (iob = .iob_ptr, binary_data = (3, record_header));
: 1916 P 2545 2 $xpo_put (iob = .iob_ptr,
: 1917 2546 2 binary_data = (CH$ALLOCATION (.txtlen), .toclob [iob$a_data]));
: 1918 2547 2
: 1919 2548 2 END;
: 1920 2549 2
: 1921 2550 1 END;

```

IOB\$=
IOB\$= TOCLOB
TOCLOB

01FC 00000 PROCESS_ENTITY_TXT:

58	00000000G	EF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8	
57	00000000G	EF	9E	00009	MOVAB	XPOS\$PUT, R8	
56	00000000G	EF	9E	00010	MOVAB	XPOS\$GET, R7	
55	00000000G	EF	9E	00017	MOVAB	MINOR, R6	
54	00000000G	EF	9E	0001E	MOVAB	TXLEN, R5	
53	00000000'	EF	9E	00025	MOVAB	XPOS\$FAILURE, R4	
5E		14	C2	0002C	MOVAB	IOB\$+52, R3	
	03B4	C3	D4	0002F	SUBL2	#20, SP	
50		66	D0	00033	CLRL	ERRCNT	
0B		50	D1	00036	MOVL	MINOR, R0	
		05	13	00039	CMPL	R0, #11	
0C		50	D1	0003B	BEQL	1\$	
		07	12	0003E	CMPL	R0, #12	
52	00C4	C3	9E	00040	BNEQ	2\$	
		25	11	00045	MOVAB	FGIOB, IOB_PTR	
0E		50	D1	00047	BRB	6\$	
		05	13	0004A	CMPL	R0, #14	
0F		50	D1	0004C	BEQL	3\$	
		07	12	0004F	CMPL	R0, #15	
52	01B8	C3	9E	00051	BNEQ	4\$	
		14	11	00056	MOVAB	TBIOB, IOB_PTR	
					BRB	6\$	

2468
2510
2512
2514
2515
2516
2517

	08		50	D1	00058	4\$:	CMPL	R0, #8	2518
	09		0A	13	0005B		BEQL	5\$	2519
	52		50	D1	0005D		CMPL	R0, #9	
			05	13	00060		BEQL	5\$	
	52		01	CE	00062		MNEGL	#1, IOB_PTR	
			05	11	00065		BRB	6\$	
	52	02AC	C3	9E	00067	5\$:	MOVAB	EXIOB, IOB_PTR	
	63		04	B0	0006C	6\$:	MOVW	#4, IOB\$+52	2525
02	A3		02	90	0006F		MOVB	#2, IOB\$+54	
F8	A3		06	90	00073		MOVB	#6, IOB\$+44	
			54	DD	00077		PUSHL	R4	
			7E	D4	00079		CLRL	-(SP)	
		CC	A3	9F	0007B		PUSHAB	IOB\$	
	67		03	FB	0007E		CALLS	#3, XPOSGET	
	65	04	B3	D0	00081		MOVL	@TOCIOB+56, TXTLEN	2526
08	AE	00000000G	EF	D0	00085		MOVL	MAJOR, RECORD_HEADER	2528
0C	AE		66	D0	0008D		MOVL	MINOR, RECORD_HEADER+4	2529
	50		65	D0	00091		MOVL	TXTLEN, R0	2530
10	AE		50	D0	00094		MOVL	R0, RECORD_HEADER+8	
			1C	13	00098		BEQL	7\$	2534
	50		03	C0	0009A		ADDL2	#3, R0	2536
	50		04	C6	0009D		DIVL2	#4, R0	
63	50		04	A5	000A0		MULW3	#4, R0, IOB\$+52	
	02		02	90	000A4		MOVB	#2, IOB\$+54	
F8	A3		06	90	000A8		MOVB	#6, IOB\$+44	
			54	DD	000AC		PUSHL	R4	
			7E	D4	000AE		CLRL	-(SP)	
		CC	A3	9F	000B0		PUSHAB	IOB\$	
	67		03	FB	000B3		CALLS	#3, XPOSGET	
			01	DD	000B6	7\$:	PUSHL	#1	2538
00000000V	EF		01	FB	000B8		CALLS	#1, VFYCAP	
	4A		50	E9	000BF		BLBC	R0, 8\$	
	6E		0C	B0	000C2		MOVW	#12, \$IOB\$OUTPUT	2544
02	AE		02	90	000C5		MOVB	#2, \$IOB\$OUTPUT+2	
03	AE		01	90	000C9		MOVB	#1, \$IOB\$OUTPUT+3	
04	AE	08	AE	9E	000CD		MOVAB	RECORD_HEADER, \$IOB\$OUTPUT+4	
44	A2		6E	9E	000D2		MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
2C	A2		07	90	000D6		MOVB	#7, 44(IOB_PTR)	
			54	DD	000DA		PUSHL	R4	
			7E	D4	000DC		CLRL	-(SP)	
			52	DD	000DE		PUSHL	IOB_PTR	
	68		03	FB	000E0		CALLS	#3, XPOSPUT	
50	65		03	C1	000E3		ADDL3	#3, TXLEN, R0	2546
	50		04	C6	000E7		DIVL2	#4, R0	
6E	50		04	A5	000EA		MULW3	#4, R0, \$IOB\$OUTPUT	
	02		02	90	000EE		MOVB	#2, \$IOB\$OUTPUT+2	
	03		01	90	000F2		MOVB	#1, \$IOB\$OUTPUT+3	
	04	04	A3	D0	000F6		MOVL	TOCIOB+56, \$IOB\$OUTPUT+4	
	44		6E	9E	000FB		MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
	2C		07	90	000FF		MOVB	#7, 44(IOB_PTR)	
			54	DD	00103		PUSHL	R4	
			7E	D4	00105		CLRL	-(SP)	
			52	DD	00107		PUSHL	IOB_PTR	
	68		03	FB	00109		CALLS	#3, XPOSPUT	
			04	0010C	8\$:		RET		2550

CONTENTS
V04-000

CONTENTS - Table of Contents formatter program 16-Sep-1984 00:01:05
PROCESS_ENTITY_TXT - Process figure/table/comp 14-Sep-1984 13:05:48

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CONTENTS.BLI;1

Page 79
(6)

CC
V0

```

1923 2551 1 %SBTTL 'TOCFIN - Finish table of contents processing'
1924 2552 1 GLOBAL ROUTINE tocfin =
1925 2553 1 ++
1926 2554 1
1927 2555 1 FUNCTIONAL DESCRIPTION:
1928 2556 1
1929 2557 1 This routine is called to finish up processing the table of contents.
1930 2558 1
1931 2559 1 This entails writing the figure, table and example information,
1932 2560 1 closing and deleting the temporary files and closing the output file.
1933 2561 1
1934 2562 1 FORMAL PARAMETERS:
1935 2563 1
1936 2564 1 None
1937 2565 1
1938 2566 1 IMPLICIT INPUTS:
1939 2567 1
1940 2568 1 None
1941 2569 1
1942 2570 1 IMPLICIT OUTPUTS:
1943 2571 1
1944 2572 1 None
1945 2573 1
1946 2574 1 ROUTINE VALUE:
1947 2575 1 COMPLETION CODES:
1948 2576 1
1949 2577 1 true
1950 2578 1
1951 2579 1 SIDE EFFECTS:
1952 2580 1
1953 2581 1 None
1954 2582 1 --
1955 2583 1
1956 2584 2 BEGIN
1957 2585 2
1958 2586 2 Close the caption temp files
1959 2587 2
1960 2588 2 $xpo_close (iob = exiob, options = remember);
1961 2589 2 $xpo_close (iob = fgiob, options = remember);
1962 2590 2 $xpo_close (iob = tbiob, options = remember);
1963 2591 2 $xpo_close (iob = stciob, options = remember);
1964 2592 2
1965 2593 2 Process the caption temp files
1966 2594 2
1967 2595 2 IF .captyp EQL new_captions
1968 2596 2 THEN
1969 2597 3 BEGIN
1970 2598 3
1971 2599 3 Using .HEADER X captions
1972 2600 3
1973 2601 3 IF .cmdblk [contents$v_tms11]
1974 2602 3 THEN
1975 2603 4 BEGIN
1976 2604 4
1977 2605 4 For TMS...
1978 2606 4
1979 2607 4 hdr_x (exiob, tms_ex, min_examp_inf, min_examp_txt, min_examp_txt_u);

```

```
1980 2608 4      hdr_x (fgiob, tms_fg, min_figur_inf, min_figur_txt, min_figur_txt_u);
1981 2609 4      hdr_x (tbiob, tms_tb, min_table_inf, min_table_txt, min_table_txt_u);
1982 2610 4      END
1983 2611 3      ELSE
1984 2612 4      BEGIN
1985 2613 4      For RUNOFF...
1986 2614 4      :
1987 2615 4      :
1988 2616 4      :   hdr_x (exiob, rno_ex, min_examp_inf, min_examp_txt, min_examp_txt_u);
1989 2617 4      :   hdr_x (fgiob, rno_fg, min_figur_inf, min_figur_txt, min_figur_txt_u);
1990 2618 4      :   hdr_x (tbiob, rno_tb, min_table_inf, min_table_txt, min_table_txt_u);
1991 2619 3      :   END;
1992 2620 3      :
1993 2621 2      :   END;
1994 2622 2      :
1995 2623 2      :   Process the .SEND TOC temp file
1996 2624 2      :   stchr ();
1997 2625 2      :
1998 2626 2      :   Delete temp files
1999 2627 2      :
2000 2628 2      :
2001 2629 2      :   $xpo_delete (iob = exiob);
2002 2630 2      :   $xpo_delete (iob = fgiob);
2003 2631 2      :   $xpo_delete (iob = tbiob);
2004 2632 2      :   $xpo_delete (iob = stciob);
2005 2633 2      :
2006 2634 2      :   Write appropriate epilogue
2007 2635 2      :
2008 2636 2      :   IF .cmdblk [contents$v_tmsli]
2009 2637 2      :   THEN
2010 2638 2      :       put ('*cfini*')
2011 2639 3      :   ELSE
2012 2640 3      :       put ('.RESTORE');
2013 2641 3      :
2014 2642 2      :
2015 2643 2      :   Finally, close the output file and call it quits
2016 2644 2      :
2017 2645 2      :   IF .cmdblk [contents$v_log]
2018 2646 2      :   THEN
2019 2647 2      :   THEN
2020 2648 2      :   %IF %BLISS (BLISS32)
2021 2649 2      :   %THEN
2022 2650 2      :       SIGNAL (contents$_created, 1, tocoob [iob$_resultant]);
2023 2651 2      :   %ELSE
2024 2652 2      :       $xpo_put_msg (severity = success,
2025 2653 2      :           string = $str_concat ('', tocoob [iob$_resultant], '' created'));
2026 2654 2      :   %FI
2027 2655 2      :
2028 2656 2      :   IF .cmdblk [contents$v_output]
2029 2657 2      :   THEN
2030 2658 2      :       $xpo_close (iob = tocoob)
2031 2659 3      :   ELSE
2032 2660 3      :       BEGIN
2033 2661 3      :       :
2034 2662 3      :       :   User specified /NOOUTPUT
2035 2663 3      :       :   Close and delete the output file.
2036 2664 3      :       :
```

```

: 2037
: 2038
: 2039
: 2040
: 2041
: 2042
2665 3      $xpo_close (iob = tocoob, options = remember);
2666 3      $xpo_delete (iob = tocoob);
2667 2      END;
2668 2
2669 2      RETURN true;
2670 1      END;

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2
45 2A 69 6E 69 66 63 2A 00280 P.ACX: .ASCII \*cfini*\
52 4F 54 53 45 52 2E 00287 P.ACY: .ASCII \.RESTORE\

```

```

.PSECT $OWNS,NOEXE,2
0007 00554 $STR$STRING:
01 0E 00556 .WORD 7
00000000' 00558 .BYTE 14, 1
0008 0055C $STR$STRING:
01 0E 0055E .WORD 8
00000000' 00560 .BYTE 14, 1
00000000' 00560 .ADDRESS P.ACY

```

```

IOB$= EXIOB
IOB$= FGIOB
IOB$= TBIOB
IOB$= EXIOB
IOB$= FGIOB
IOB$= TBIOB

```

.EXTRN XPOS\$DELETE

.PSECT \$CODE\$,NOWRT,2

```

OFFC 00000
5B 00000000G EF 9E 00002 MOVAB STR$FAILURE, R11
5A 00000000G EF 9E 00009 MOVAB $STR$TARGET, R10
59 00000000G EF 9E 00010 MOVAB XPOS$DELETE, R9
58 00000000G EF 9E 00017 MOVAB HDR X, R8
57 00000000G EF 9E 0001E MOVAB CMDBLK, R7
56 00000000G EF 9E 00025 MOVAB IOB$+44, R6
55 00000000G EF 9E 0002C MOVAB XPOS$CLOSE, R5
54 00000000G EF 9E 00033 MOVAB IOB$+44, R4
53 00000000G EF 9E 0003A MOVAB XPOS$FAILURE, R3
52 00000000' EF 9E 00041 MOVAB IOB$, R2
2E A2 10 88 00048 BISB2 #16, IOB$+46
2C A2 02 90 0004C MOVAB #2, IOB$+44
53 DD 00050 PUSHL R3
7E D4 00052 CLRL -(SP)
52 DD 00054 PUSHL R2
03 FB 00056 CALLS #3, XPOS$CLOSE
FE46 C2 10 88 00059 BISB2 #16, IOB$+46
FE44 C2 02 90 0005E MOVAB #2, IOB$+44
53 DD 00063 PUSHL R3
7E D4 00065 CLRL -(SP)

```

2588

2589

			FE18	C2	9F	00067	PUSHAB	IOB\$		
				03	FB	00068	CALLS	#3, XPOS\$CLOSE		
	FF3A	65		10	88	0006E	BISB2	#16, IOB\$+46	2590	
	FF38	C2		02	90	00073	MOV8	#2, IOB\$+44		
				53	DD	00078	PUSHL	R3		
				7E	D4	0007A	CLRL	-(SP)		
			FF0C	C2	9F	0007C	PUSHAB	IOB\$		
		65		03	FB	00080	CALLS	#3, XPOS\$CLOSE		
	02	A6		10	88	00083	BISB2	#16, IOB\$+46	2591	
		66		02	90	00087	MOV8	#2, IOB\$+44		
				53	DD	0008A	PUSHL	R3		
				7E	D4	0008C	CLRL	-(SP)		
			D4	A6	9F	0008E	PUSHAB	IOB\$		
		65		03	FB	00091	CALLS	#3, XPOS\$CLOSE		
		01	0100	C2	D1	00094	CMPL	CAPTYP, #1	2595	
				6D	12	00099	BNEQ	3\$		
32		67		01	E1	0009B	BBC	#1, CMDBLK, 1\$	2601	
				09	DD	0009F	PUSHL	#9	2607	
				08	DD	000A1	PUSHL	#8		
				07	DD	000A3	PUSHL	#7		
			00000000G	EF	9F	000A5	PUSHAB	TMS_EX		
				52	DD	000AB	PUSHL	R2		
		68		05	FB	000AD	CALLS	#5, HDR_X		
				0C	DD	000B0	PUSHL	#12	2608	
				0B	DD	000B2	PUSHL	#11		
				0A	DD	000B4	PUSHL	#10		
			00000000G	EF	9F	000B6	PUSHAB	TMS_FG		
			FE18	C2	9F	000BC	PUSHAB	FGIOB		
		68		05	FB	000C0	CALLS	#5, HDR_X		
				0F	DD	000C3	PUSHL	#15	2609	
				0E	DD	000C5	PUSHL	#14		
				0D	DD	000C7	PUSHL	#13		
			00000000G	EF	9F	000C9	PUSHAB	TMS_TB		
				30	11	000CF	BRB	2\$		
				09	DD	000D1	PUSHL	#9	2616	
				08	DD	000D3	PUSHL	#8		
				07	DD	000D5	PUSHL	#7		
			00000000G	EF	9F	000D7	PUSHAB	RNO_EX		
				52	DD	000DD	PUSHL	R2		
		68		05	FB	000DF	CALLS	#5, HDR_X		
				0C	DD	000E2	PUSHL	#12	2617	
				0B	DD	000E4	PUSHL	#11		
				0A	DD	000E6	PUSHL	#10		
			00000000G	EF	9F	000E8	PUSHAB	RNO_FG		
			FE18	C2	9F	000EE	PUSHAB	FGIOB		
		68		05	FB	000F2	CALLS	#5, HDR_X		
				0F	DD	000F5	PUSHL	#15	2618	
				0E	DD	000F7	PUSHL	#14		
				0D	DD	000F9	PUSHL	#13		
			00000000G	EF	9F	000FB	PUSHAB	RNO_TB		
			FF0C	C2	9F	00101	PUSHAB	TBIOB		
				05	FB	00105	CALLS	#5, HDR_X		
				00	FB	00108	CALLS	#0, STCRDR	2626	
				03	90	0010F	MOV8	#3, IOB\$+44	2630	
				53	DD	00113	PUSHL	R3		
				7E	D4	00115	CLRL	-(SP)		
				52	DD	00117	PUSHL	R2		
	00000000G	68								
		EF								
	2C	A2								

	FE44	69 C2			03 FB 00119 03 90 0011C 53 DD 00121 7E D4 00123 C2 9F 00125 03 FB 00129 03 90 0012C 53 DD 00131 7E D4 00133 C2 9F 00135 03 FB 00139 03 90 0013C 53 DD 0013F 7E D4 00141 A6 9F 00143 03 FB 00146 01 E1 00149 5B DD 0014D 7E D4 0014F 5A DD 00151 C2 9F 00153 0A 11 00157 5B DD 00159 4\$: 7E D4 0015B 5A DD 0015D C2 9F 0015F 7E D4 00163 5\$: 05 FB 00165 6A 3C 0016C 50 C0 0016F 6A 9E 00176 07 90 0017A 53 DD 0017D 7E D4 0017F D4 A4 9F 00181 03 FB 00184 01 E1 0018B 00 FB 0018F 05 E1 00196 6\$: A4 9F 0C19A 01 DD 0019D 8F DD 0019F 03 FB 001A5 67 E9 001AC 7\$: 02 90 001AF 53 DD 001B2 7E D4 001B4 D4 A4 9F 001B6 03 FB 001B9 1E 11 001BC 1A 88 001BE 8\$: 0C 90 001C2 53 DD 001C5 7E D4 001C7 D4 A4 9F 001C9 03 FB 001CC 03 90 001CF	CALLS #3, XPOSDELETE MOVB #3, IOB\$+44 PUSHL R3 CLRL -(SP) PUSHAB IOB\$ CALLS #3, XPOSDELETE MOVB #3, IOB\$+44 PUSHL R3 CLRL -(SP) PUSHAB IOB\$ CALLS #3, XPOSDELETE MOVB #3, IOB\$+44 PUSHL R3 CLRL -(SP) PUSHAB IOB\$ CALLS #3, XPOSDELETE BBC #1, CMDBLK, 4\$ PUSHL R11 CLRL -(SP) PUSHL R10 PUSHAB \$\$? ? KING BRB 5\$ PUSHL R11 CLRL -(SP) PUSHL R10 PUSHAB \$STR\$STRING CLRL -(SP) CALLS #5, XST\$COPY MOVZWL TMPSTR, R0 ADDL2 R0, CHROUT MOVAB \$IOB\$OUTPUT, IOB\$+68 MOVB #7, IOB\$+44 PUSHL R3 CLRL -(SP) PUSHAB IOB\$ CALLS #3, XPOS\$PUT BBC #1, CMDBLK, 6\$ CALLS #0, SPLIT BBC #5, CMDBLK, 7\$ PUSHAB TOC00B+28 PUSHL #1 PUSHL #DSRTOC\$ CREATED CALLS #3, LIB\$SIGNAL BLBC CMDBLK, 8\$ MOVB #2, IOB\$+44 PUSHL R3 CLRL -(SP) PUSHAB IOB\$ CALLS #3, XPOS\$CLOSE BRB 9\$ BISB2 #16, IOB\$+46 MOVB #2, IOB\$+44 PUSHL R3 CLRL -(SP) PUSHAB IOB\$ CALLS #3, XPOS\$CLOSE MOVB #3, IOB\$+44	2631 2632 2633 2637 2639 2641 2646 2650 2656 2658 2665 2666
--	------	----------	--	--	---	--	--

CONTENTS - Table of Contents formatter program
TOCFIN - Finish table of contents processing

B 14
16-Sep-1984 00:01:05
14-Sep-1984 13:05:48

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CONTENTS.BLI;1

Page 85
(7)

COM
VO4

```

69          D4      53 DD 001D2      PUSHL R3
50          7E      D4 001D4      CLRL -(SP)
          A4      9F 001D6      PUSHAB IOBS
          03      FB 001D9      CALLS #3, XPOS$DELETE
          01      D0 001DC 9$:      MOVL #1, R0
          04      001DF      RET

```

```
; Routine Size: 480 bytes,    Routine Base: $CODE$ + 1F19
```

```

2044 2671 1 %SBTTL 'VFYCAP - Check caption type'
2045 2672 1 ROUTINE vfyCAP (type) =
2046 2673 1 ++
2047 2674 1
2048 2675 1 FUNCTIONAL DESCRIPTION:
2049 2676 1
2050 2677 1 This routine checks the caption type being processed against the
2051 2678 1 argument caption type. If they are the same, a value of true is
2052 2679 1 returned.
2053 2680 1
2054 2681 1 If they are different, a message is written if no message has been
2055 2682 1 written already, and false is returned.
2056 2683 1
2057 2684 1 If no caption type has been set yet, the caption type is set from
2058 2685 1 the argument type and true is returned.
2059 2686 1
2060 2687 1 FORMAL PARAMETERS:
2061 2688 1
2062 2689 1 type - argument caption type
2063 2690 1
2064 2691 1 IMPLICIT INPUTS:
2065 2692 1
2066 2693 1 captyp - caption type being processed
2067 2694 1 capmsg - true if a message has been written
2068 2695 1
2069 2696 1 IMPLICIT OUTPUTS:
2070 2697 1
2071 2698 1 captyp - will be set if it is zero
2072 2699 1 capmsg - will be set to true if a message is written
2073 2700 1
2074 2701 1 ROUTINE VALUE:
2075 2702 1 COMPLETION CODES:
2076 2703 1
2077 2704 1 See functional description above.
2078 2705 1
2079 2706 1 SIDE EFFECTS:
2080 2707 1
2081 2708 1 None
2082 2709 1 --
2083 2710 1
2084 2711 2 BEGIN
2085 2712 2
2086 2713 2 SELECTONE .captyp OF
2087 2714 2 SET
2088 2715 2
2089 2716 2 [old_captions] :
2090 2717 2
2091 2718 2 IF .type EQL new_captions
2092 2719 2 THEN
2093 2720 3 BEGIN
2094 2721 3
2095 2722 3 IF NOT .capmsg
2096 2723 3 THEN
2097 2724 4 BEGIN
2098 2725 4 capmsg = true;
2099 2726 4
2100 L 2727 4 %IF %BLISS (BLISS32)

```



```

: 2101      2728 4 %THEN
: 2102      2729 4          SIGNAL (contents$_captions, 0, contents$_ignorenew);
: 2103      2730 4 %ELSE
: 2104      2731 4          $xpo_put_msg (severity = warning,
: 2105      2732 4          string = 'both .SEND TOC captions and ".HEADER x" captions encountered',
: 2106      2733 4          string = '".HEADER x" captions will be ignored');
: 2107      2734 4 %FI
: 2108      2735 4
: 2109      2736 3          END;
: 2110      2737 3
: 2111      2738 3          RETURN false;
: 2112      2739 2          END;
: 2113      2740 2
: 2114      2741 2          [new_captions] :
: 2115      2742 2
: 2116      2743 2          IF .type EQL old_captions
: 2117      2744 2          THEN
: 2118      2745 3              BEGIN
: 2119      2746 3
: 2120      2747 3          IF NOT .capmsg
: 2121      2748 3          THEN
: 2122      2749 4              BEGIN
: 2123      2750 4                  capmsg = true;
: 2124      2751 4
: 2125      2752 4 %IF %BLISS (BLISS32)
: 2126      2753 4 %THEN
: 2127      2754 4          SIGNAL (contents$_captions, 0, contents$_ignoreold);
: 2128      2755 4 %ELSE
: 2129      2756 4          $xpo_put_msg (severity = warning,
: 2130      2757 4          string = 'both .SEND TOC captions and ".HEADER x" captions encountered',
: 2131      2758 4          string = '.SEND TOC captions will be ignored');
: 2132      2759 4 %FI
: 2133      2760 4
: 2134      2761 3          END;
: 2135      2762 3
: 2136      2763 3          RETURN false;
: 2137      2764 2          END;
: 2138      2765 2
: 2139      2766 2          [OTHERWISE] :
: 2140      2767 2              captyp = .type;
: 2141      2768 2
: 2142      2769 2          TES;
: 2143      2770 2
: 2144      2771 2          RETURN true;
: 2145      2772 1          END;

```

			0004 0000	VFYCAP: .WORD	Save R2	
52	00000000'	EF	9E 00002	MOVAB	CAPMSG, R2	: 2672
50	FC	A2	D0 00009	MOVL	CAPTYP, R0	: 2713
FFFFFFF	8F	50	D1 0000D	CMPL	R0, #-1	: 2716
		14	12 00014	BNEQ	1\$	
01	04	AC	D1 00016	CMPL	TYPE, #1	: 2718
		3F	12 0001A	BNEQ	4\$	

CONTENTS
V04-000

CONTENTS - Table of Contents formatter program
VFYCAP - Check caption type

E 14
16-Sep-1984 00:01:05
14-Sep-1984 13:05:48

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CONTENTS.BLI;1

Page 88
(8)

40	62	E8	0001C	BLBS	CAPMSG, 5\$: 2722
62	01	D0	0001F	MOVL	#1, CAPMSG	: 2725
	00000000G	8F	DD 00022	PUSHL	#DSRTOCS_IGNORENEW	: 2729
		1B	11 00028	BRB	2\$: 2741
01		50	D1 0002A 1\$:	CMPL	R0, #1	: 2743
		27	12 0002D	BNEQ	3\$: 2747
FFFFFFF	8F	04	AC D1 0002F	CMPL	TYPE, #-1	: 2750
		22	12 00037	BNEQ	4\$: 2754
23		62	E8 00039	BLBS	CAPMSG, 5\$: 2763
62		01	D0 0003C	MOVL	#1, CAPMSG	: 2767
	00000000G	8F	DD 0003F	PUSHL	#DSRTOCS_IGNOREOLD	: 2771
		7E	D4 00045 2\$:	CLRL	-(SP)	: 2772
	00000000G	8F	DD 00047	PUSHL	#DSRTOCS_CAPTIONS	: 2772
00000000G	00	03	FB 0004D	CALLS	#3, LIB\$SIGNAL	: 2772
		09	11 00054	BRB	5\$: 2772
FC	A2	04	AC D0 00056 3\$:	MOVL	TYPE, CAPTYP	: 2772
	50	01	D0 0005B 4\$:	MOVL	#1, R0	: 2772
			04 0005E	RET		: 2772
		50	D4 0005F 5\$:	CLRL	R0	: 2772
			04 00061	RET		: 2772

; Routine Size: 98 bytes, Routine Base: \$CODE\$ + 20F9

```

2147 2773 1 %SBTTL 'OUTSEC - Format a section number'
2148 2774 1 ROUTINE outsec : NOVALUE =
2149 2775 1 ++
2150 2776 1 FUNCTIONAL DESCRIPTION:
2151 2777 1 OUTSEC generates a section number of the form 1.2.3.....
2152 2778 1
2153 2779 1 FORMAL PARAMETERS:
2154 2780 1
2155 2781 1 None
2156 2782 1
2157 2783 1 IMPLICIT INPUTS:
2158 2784 1
2159 2785 1 cmdblk - command line information block
2160 2786 1 lasthl - previous HL 'n'
2161 2787 1 hl_n - current HL 'n'
2162 2788 1 minor - minor record type code
2163 2789 1 pagen - current page number
2164 2790 1 hlhist - header level counters
2165 2791 1 hldsp - header level display codes
2166 2792 1 - output line variables
2167 2793 1
2168 2794 1 IMPLICIT OUTPUTS:
2169 2795 1
2170 2796 1 ptrsct - modified
2171 2797 1 lensct - length of section number
2172 2798 1 txtsct - contains formatted section number
2173 2799 1 - output line variables modified (since section number is
2174 2800 1 inserted into the output line)
2175 2801 1
2176 2802 1 ROUTINE VALUE:
2177 2803 1 COMPLETION CODES:
2178 2804 1
2179 2805 1 None
2180 2806 1
2181 2807 1 SIDE EFFECTS:
2182 2808 1
2183 2809 1 None
2184 2810 1
2185 2811 1 --
2186 2812 1
2187 2813 1 BEGIN
2188 2814 2
2189 2815 2 LOCAL
2190 2816 2 hl_width; ! Number of columns reserved for header level.
2191 2817 2
2192 2818 2 lensct = 0;
2193 2819 2 ptrsct = CH$PTR (txtsct);
2194 2820 2
2195 2821 2 IF .cmdblk [contents$v_tms11]
2196 2822 2 THEN
2197 2823 2 BEGIN
2198 2824 3
2199 2825 3 ! Write out the HL level for TMS
2200 2826 3
2201 2827 3 IF (.lasthl NEQ .hl_n) AND (.lasthl NEQ 0) THEN put ('[va5]');
2202 2828 3
2203 2829 3

```

```

: 2204      2830      3      literal_text ('*c');
: 2205      2831      3      lensct = pacbas (.hl_n, ptrsct, 10);
: 2206      2832      3      ptrsct = CHSPTR (txtsct);
: 2207      2833      3      CHSMOVE (.lensct, .ptrsct, .lp);
: 2208      2834      3      lp = CHSPLUS (.lp, .lensct);
: 2209      2835      3      intlin = .intlin + .lensct;
: 2210      2836      3      lensct = 0;
: 2211      2837      3      literal_text ('n*');
: 2212      2838      2      END;
: 2213      2839      2
: 2214      2840      2      IF .cmdblk [contents$v_include_sections]
: 2215      2841      3      AND (
: 2216      2842      4      (.cmdblk [contents$g_sections] GEQ .hl_n)
: 2217      2843      4      OR (
: 2218      2844      5      (.cmdblk [contents$g_sections] EQL -1)
: 2219      2845      5      AND (.minor NEQ min_hl_txt_u)
: 2220      2846      4      )
: 2221      2847      3      )
: 2222      2848      2      THEN
: 2223      2849      3      BEGIN
: 2224      2850      3
: 2225      2851      3      : Don't write the number if user said heads are unnumbered.
: 2226      2852      3      : The user must have requested numbered heads and
: 2227      2853      3      : the header level must be less than the largest one the user
: 2228      2854      3      : specified or the heads must be displayed as input and the
: 2229      2855      3      : text is not for an unnumbered header.
: 2230      2856      3
: 2231      2857      4      IF (.pagen [sct_ttyp] EQL 0) AND (.hl_n EQL 1)
: 2232      2858      3      THEN
: 2233      2859      4      BEGIN
: 2234      2860      4
: 2235      2861      4      : This is a HL 1 for a document not divided into sections
: 2236      2862      4
: 2237      2863      4      lensct = pacxxx (.hllist [0], ptrsct, .hldsp [0]);
: 2238      2864      4      END
: 2239      2865      3      ELSE
: 2240      2866      4      BEGIN
: 2241      2867      4
: 2242      2868      4      IF .pagen [sct_ttyp] NEQ 0
: 2243      2869      4      THEN
: 2244      2870      5      BEGIN
: 2245      2871      5
: 2246      2872      5      : Document is divided into sections
: 2247      2873      5
: 2248      2874      5      lensct = pacsec (pagen, ptrsct);
: 2249      2875      5      CHSWCHAR_A (%C'.', ptrsct);
: 2250      2876      5      lensct = .lensct + 1;
: 2251      2877      4      END;
: 2252      2878      4
: 2253      2879      4      INCR i FROM 1 TO .hl_n DO
: 2254      2880      5      BEGIN
: 2255      2881      5      lensct = .lensct + pacxxx (.hllist [.i - 1], ptrsct, .hldsp [.i - 1]);
: 2256      2882      5      CHSWCHAR_A (%C'.', ptrsct);
: 2257      2883      5      lensct = .lensct + 1;
: 2258      2884      4      END;
: 2259      2885      4
: 2260      2886      4

```

```

2261 2887 4      ! Always get an extra dot, so uncount it.
2262 2888 4
2263 2889 4      lensct = .lensct - 1;
2264 2890      END;
2265 2891
2266 2892
2267 2893      ! Now copy the converted section number into the line that's being built up.
2268 2894      fmttxt (.lensct, CH$PTR (txtsct));
2269 2895
2270 2896      IF NOT .cmdblk [contents$v_tms11] THEN write_char (' ', counts_visually);
2271 2897      END;
2272 2898
2273 2899      IF .cmdblk [contents$v_tms11]
2274 2900      THEN
2275 2901          write_char ('a')
2276 2902      ELSE
2277 2903          BEGIN
2278 2904              For RUNOFF...
2279 2905
2280 2906              Compute the width of the region into which the header level
2281 2907              number was supposed to fit.
2282 2908
2283 2909              hl_width = 0;
2284 2910
2285 2911              INCR i FROM 0 TO .hl_n DO
2286 2912                  hl_width = .hl_width + .indents [.i];
2287 2913
2288 2914
2289 2915              Pad out this .HL number to fill the columns allocated.
2290 2916
2291 2917              pad ((.hl_width - .extlin));
2292 2918          END;
2293 2919
2294 2920      END;
2295 2921
2296 2922      ! End of outsec

```

```

.PSECT $PLITS,NOWRT,NOEXE,2
5D 35 61 76 5B 0028F P.ACZ: .ASCII \[va5]\
00 00 63 2A 00294 P.ADA: .ASCII \*c\<0><0>
00 00 2A 6E 00298 P.ADB: .ASCII \n*\<0><0>

```

```

.PSECT $OWNS,NOEXE,2
0005 00564 $STR$STRING:
01 0E 00566 .WORD 5
00000000' 00568 .BYTE 14, 1
.ADDRESS P.ACZ

```

```

.PSECT $CODE$,NOWRT,2
OFFC 00000 OUTSEC: .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11

```

		5B 00000000G	EF	9E 00002	MOVAB	HL_N, R11	
		5A 00000000G	EF	9E 00009	MOVAB	CMDBLK, R10	
		59 00000000G	EF	9E 00010	MOVAB	LENSCT, R9	
		58 00000000G	EF	9E 00017	MOVAB	PTRSCT, R8	
		57 00000000G	EF	9E 0001E	MOVAB	LP, R7	
			69	D4 00025	CLRL	LENSCT	2819
03		68 00000000G	EF	9E 00027	MOVAB	TXTSCT, PTRSCT	2820
		6A	01	E0 0002E	BBS	#1, CMDBLK, 1\$	2822
			00BE	31 00032	BRW	3\$	
		6B 00000000'	EF	D1 00035	CMPL	LASTHL, HL_N	2828
			65	13 0003C	BEQL	2\$	
		00000000'	EF	D5 0003E	TSTL	LASTHL	
			5D	13 00044	BEQL	2\$	
		00000000G	EF	9F 00046	PUSHAB	STR\$FAILURE	
			7E	D4 0004C	CLRL	-(SP)	
		00000000G	EF	9F 0004E	PUSHAB	\$STR\$TARGET	
		00000000'	EF	9F 00054	PUSHAB	\$STR\$STRING	
			7E	D4 0005A	CLRL	-(SP)	
		00000000G	EF	05	FB 0005C	CALLS	#5, XST\$COPY
		50 00000000G	EF	3C 00063	MOVZWL	TMPSTR, R0	
		00000000G	EF	50	C0 0006A	ADDL2	R0, CHROUT
		00000000G	EF	9E 00071	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	EF	07	90 0007C	MOVAB	#7, IOB\$+44
				9F 00083	PUSHAB	XPOS\$FAILURE	
		00000000G	7E	D4 00089	CLRL	-(SP)	
			EF	9F 0008B	PUSHAB	IOB\$	
		00000000G	03	FB 00091	CALLS	#3, XPOS\$PUT	
07		6A	01	E1 00098	BBS	#1, CMDBLK, 2\$	
		00000000G	EF	00	FB 0009C	CALLS	#0, SPLIT
		00	B7	00000000'	EF	B0 000A3	2\$: MOVW
			67	02	C0 000AB	ADDL2	P.ADA, @LP
		00000000G	EF	02	C0 000AE	ADDL2	#2, LP
				0A	DD 000B5	PUSHL	#2, INTLIN
				58	DD 000B7	PUSHL	#10
				6B	DD 000B9	PUSHL	R8
		00000000G	EF	03	FB 000BB	PUSHL	HL_N
		69	50	D0 000C2	CALLS	#3, PACBAS	
		68 00000000G	EF	9E 000C5	MOVL	R0, LENSCT	2832
		56	69	D0 000CC	MOVAB	TXTSCT, PTRSCT	2833
		88	56	28 000CF	MOVL	LENSCT, R6	
00	B7	00	67	56	C0 000D5	MOVAB	R6, @PTRSCT, @LP
		00000000G	EF	56	C0 000D8	ADDL2	R6, LP
				69	D4 000DF	ADDL2	R6, INTLIN
		00	B7	00000000'	EF	B0 000E1	CLRL
		67	02	C0 000E9	MOVW	P.ADB, @LP	LENSCT
		00000000G	EF	02	C0 000EC	ADDL2	#2, LP
03		6A	04	E0 000F3	ADDL2	#2, INTLIN	
			00B5	31 000F7	BBS	#4, CMDBLK, 5\$	2836
		6B	14	AA	D1 000FA	BRW	12\$
				13	18 000FE	CMPL	CMDBLK+20, HL_N
		FFFFFFF	8F	14	AA	D1 00100	6\$: BGEQ
				ED	12 00108	CMPL	CMDBLK+20, #-1
		06 00000000G	EF	D1 0010A	BNEQ	4\$	2844
			E4	13 00111	CMPL	MINOR, #6	2845
			00	EF 00113	BEQL	4\$	
50		00000000G	EF	04	1F 12 0011C	EXTZV	#0, #4, PAGEN, R0
				01	6B D1 0011E	BNEQ	7\$
						CMPL	HL_N, #1
							2857

			00000000G	1A	12	00121	BNEQ	7\$		
				EF	DD	00123	PUSHL	HLDSP		2863
			00000000G	58	DD	00129	PUSHL	R8		
				EF	DD	00128	PUSHL	HLLIST		
		00000000G	EF	03	FB	00131	CALLS	#3, PACXXX		
		69		50	D0	00138	MOVL	R0, LENSCT		
				4D	11	00138	BRB	11\$		2857
				50	D5	0013D	TSTL	R0		2868
				1A	13	0013F	BEQL	8\$		
				58	DD	00141	PUSHL	R8		2874
			00000000G	EF	9F	00143	PUSHAB	PAGEN		
		00000000G	EF	02	FB	00149	CALLS	#2, PACSEC		
		69		50	D0	00150	MOVL	R0, LENSCT		
		00	B8	2E	90	00153	MOVB	#46, @PTRSCT		2875
				68	D6	00157	INCL	PTRSCT		
				69	D6	00159	INCL	LENSCT		2876
			53	6B	D0	0015B	MOVL	HL_N, R3		2879
				52	D4	0015E	CLRL	I		
				22	11	00160	BRB	10\$		
			00000000G	EF	42	DD	00162	PUSHL	HLDSP-4[I]	2881
				58	DD	00169	PUSHL	R8		
			00000000G	EF	42	DD	0016B	PUSHL	HLLIST-4[I]	
		00000000G	EF	03	FB	00172	CALLS	#3, PACXXX		
		69		50	C0	00179	ADDL2	R0, LENSCT		
		00	B8	2E	90	0017C	MOVB	#46, @PTRSCT		2882
				68	D6	00180	INCL	PTRSCT		
				69	D6	00182	INCL	LENSCT		2883
DA			52	53	F3	00184	AOBLEQ	R3, I, 9\$		2879
				69	D7	00188	DECL	LENSCT		2889
			00000000G	EF	9F	0018A	PUSHAB	TXTSCT		2895
				69	DD	00190	PUSHL	LENSCT		
			00000000G	EF	02	FB	00192	CALLS	#2, FMTTXT	
16			6A	01	E0	00199	BBS	#1, CMDBLK, 13\$		2897
		00	B7	20	90	0019D	MOVB	#32, @LP		
				67	D6	001A1	INCL	LP		
			00000000G	EF	D6	001A3	INCL	INTLIN		
			00000000G	EF	D6	001A9	INCL	EXTLIN		
OE			6A	01	E1	001AF	BBC	#1, CMDBLK, 14\$		2900
		00	B7	40	8F	90	001B3	MOVB	#64, @LP	2902
				67	D6	001B8	INCL	LP		
			00000000G	EF	D6	001BA	INCL	INTLIN		
				04	001C0	RET				2900
				50	D4	001C1	CLRL	HL_WIDTH		2911
			51	01	CE	001C3	MNEGL	#1, I		2913
				05	11	001C6	BRB	16\$		
			50	20	AA	41	C0	001C8	15\$:	2914
F7			51	6B	F3	001CD	AOBLEQ	HL_N, I, 15\$		
		00000000G	EF	50	D1	001D1	CMPL	HL_WIDTH, EXTLIN		2919
				20	15	001D8	BLEQ	17\$		
56			50	EF	C3	001DA	SUBL3	EXTLIN, HL_WIDTH, R6		
20			6E	00	2C	001E2	MOVCS	#0, (SP), #32, R6, @LP		
				00	B7	001E7				
				67	56	C0	001E9	ADDL2	R6, LP	
			00000000G	EF	56	C0	001EC	ADDL2	R6, INTLIN	
			00000000G	EF	56	C0	001F3	ADDL2	R6, EXTLIN	
					04	001FA	17\$:	RET		2922

: Routine Size: 507 bytes, Routine Base: \$CODE\$ + 2158

: 2297 2923 1
: 2298 2924 1 END ! End of module
: 2299 2925 0 ELUDGM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes					
\$OWN\$	1388	NOVEC,	WRT,	RD	NOEXE,NOSHR,	LCL, REL,	CON,NOPI,ALIGN(2)
\$PLITS	668	NOVEC,NOWRT,	RD	NOEXE,NOSHR,	LCL, REL,	CON,NOPI,ALIGN(2)	
\$CODE\$	9046	NOVEC,NOWRT,	RD	EXE,NOSHR,	LCL, REL,	CON,NOPI,ALIGN(2)	

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	219 37	252	00:00.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:CONTENTS/OBJ=OBJ\$:CONTENTS MSRC\$:CONTENTS/UPDATE=(ENHS:CONTENTS)

: Size: 9046 code + 2056 data bytes
: Run Time: 04:38.3
: Elapsed Time: 08:52.8
: Lines/CPU Min: 630
: Lexemes/CPU-Min: 91165
: Memory Used: 1086 pages
: Compilation Complete

0338 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

